Einfach Kodéieren

LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de l'Éducation nationale,
de l'Enfance et de la Jeunesse

Einfach Kodéieren

SCRIPT
Service de Coordination de la Recherche
et de l'Innovation pédagogiques et technologiques

einfach digital
Zukunftskompetenze
fir staark Kanner

Titre : **Einfach Kodéieren**

Éditeur : SCRIPT, Service de Coordination de la Recherche et de l'Innovation pédagogiques et technologiques

eduPôle Walferdange
28, route de Diekirch
L-7220 Walferdange
Tél. : 247-85187
secretariat@script.lu
www.script.lu

Réalisation/Conception : SCRIPT, Service de Coordination de la Recherche et de l'Innovation pédagogiques et technologiques

Numéro interne : 2020SCRIPTlivre001

SCRIPT, Walferdange 2020

LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de l'Éducation nationale,
de l'Enfance et de la Jeunesse

Inhalts-verzeichnis/

Einführung

- Einleitung 5
- Was bedeuten: Computational Thinking, mathematisches Denken und Kodieren? 6
- Die acht Kompetenzen des Computational Thinking 8
- Umsetzung des Computational Thinking im Unterricht 10
- Inhalt des Starter-Kits 13
- Erklärungen zu den Unterrichtsideen 14

Glossar 16

Ozobot Farbcodes

- Flächeninhalte und Umfänge 21
- Challenge Cards 22
- Ozobot Kopfrechnen 28
- Interaktive Geschichte 34
- Würfel Challenge 36

Ozobot Ozoblocky

- Shapetracer 38
- Labyrinth entwerfen 40
- Slotcar 42
- Vielecke 45
- Stadtplan 46

Scratch

- Koordinatensystem 50
- Maussteuerung und Katzenbilder 52
- Nachrichten versenden 54
- Labyrinth 55

Kniwwelino

- Stoppuhr 59
- Happy Birthday 61
- Kniwwelino Matrixanzeige 65
- Textnachrichten-Armband entwerfen 67
- Stein-Papier-Schere 69

Modi

- Taschenlampe 72
- Geräuschepegel 73
- Polizei 74
- Instrument 76
- Rover 78

Quellen 81

Weiterführende Unterrichtsideen 82

Kontakt 84

Einführung

Einleitung/

In unserer Welt übernehmen Computer immer mehr die Aufgabe, Informationen und Daten zu produzieren und zu verwalten. Oftmals ganz automatisch und ohne direktes Eingreifen des Menschen. Für diese Prozesse basieren sich Computer auf genaue Handlungsanweisungen, also auf eine bestimmte Anzahl an Aktionen, welche in einer festgelegten Reihenfolge durchgeführt werden müssen. Diese Handlungsanweisungen werden Algorithmen genannt.

Algorithmen prägen unseren Alltag in immer stärkerem Ausmaß: in der virtuellen Welt des Internets durch persönliche Produktempfehlungen aufgrund von Daten, die beim Kauf mit einer Kundenkarte gesammelt werden oder durch frühzeitige Erkennung von Krankheitssymptomen in der Medizin. Algorithmen arbeiten schnell und effektiv im Hintergrund und treffen auch Entscheidungen für uns und über uns. Aus diesem Grund wird das Verständnis über die Wirkungsweise der Algorithmen zu einem wichtigen Bestandteil der Bildung in unserer digitalen Gesellschaft. Das Bewusstsein für Medienbildung und Mediennutzung ist in den Grundschulen angekommen; das Erlernen und die Anwendung von Algorithmen werden in den kommenden Jahren dieses Bewusstsein ergänzen. Indem wir junge Menschen für den Blick unter die Oberfläche sensibilisieren und auf diese Weise ihr Interesse wecken, bauen wir gleichzeitig auch wichtige Grundkenntnisse über digitale Geräte auf.

Für die SchülerInnen der Grundschulen bedeutet diese Kompetenzerweiterung, die Technik eines Gerätes mit seinen Funktionen und Mechanismen, nach denen sie funktioniert, kennenzulernen und verstehen zu können. Dieses Verständnis hilft jungen Menschen kritikfähig, gegenüber den medialen Realitäten und ihren Entstehungsprozessen zu sein und sich aktiv am Aufbau der eigenen Umwelt zu beteiligen.

Was bedeuten: Computational Thinking, mathematisches Denken und Kodieren?/

Der Begriff Computational Thinking *(informatisches Denken)* steht für:

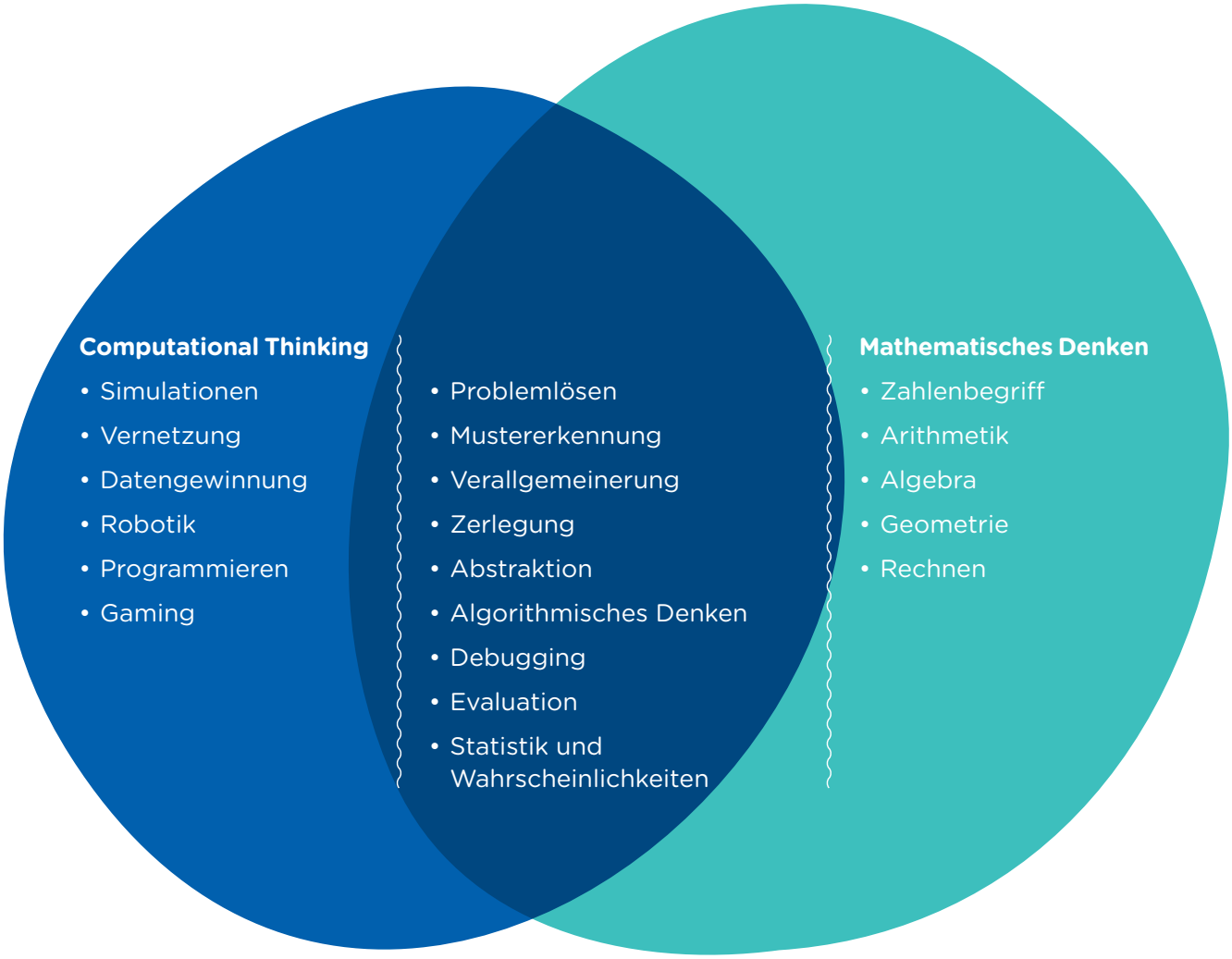
1. einen Gedankenprozess,

2. der zu einer Lösung eines Problems führen soll und

3. so notiert ist, dass ein Mensch oder ein Computer/Roboter diese Lösung ausführen kann.

(Wing, 2010)

Computational Thinking ist demnach ein kognitiver Prozess zum Problemlösen *(problemlösendes Denken)*, der sich in vielen Punkten mit dem mathematischen Denken überschneidet. Die folgende Abbildung zeigt die Gemeinsamkeiten zwischen beiden Prozessen auf. Für die SchülerInnen schließt das Erlernen von mathematischem Denken gleichzeitig eine Reihe von Kompetenzen des Computational Thinking mit ein.



Überschneidungen vom Computational Thinking und mathematischem Denken *(nach Shute et al., 2017, S. 146)*

Der Begriff **Kodieren** kann gleichgesetzt werden mit dem deutschen Begriff Programmieren. Programmieren bedeutet die Beschreibung einer Handlungsabfolge in einer für eine Maschine verständlichen Art und Weise. Kodieren konzentriert sich auf eine geeignete Darstellung der Lösungen von Problemen, entweder als textbasierte oder als grafische Aufbereitung. Für diese Vorgehensweise ist zunächst eine gründliche Analyse des Problems erforderlich, der eine sorgfältige und präzise Bearbeitung folgt, mit einer anschließenden Darstellung der Lösung. Erst dann können Algorithmen eingesetzt werden, welche durch Kodieren in eine für Maschinen verständliche Sprache *(Programmcode)* notiert werden. Somit stellen die Kompetenzen des Computational Thinking und des mathematischen Denkens die Grundlage für das Kodieren dar.

Die acht Kompetenzen des Computational Thinking/

Zum Fördern des problemlösenden Denkens bedarf es grundlegender Kompetenzen. **Computational Thinking** kann anhand folgender acht Kompetenzen definiert werden. Diese liegen auch dem mathematischen Denken zu Grunde.

	Mustererkennung	*Mustererkennung bedeutet, in einer Menge von Daten Regelmäßigkeiten, Wiederholungen und Ähnlichkeiten zu erkennen und daraus Regeln zu erstellen, wie allgemeine Probleme gelöst werden können.*
	Verallgemeinerung oder Generalisierung	*Verallgemeinerung ist die Fähigkeit, bereits bekannte Probleme und Lösungen zu erkennen und sie in neuen Szenarien und Kontexten wiederzuverwenden oder neu anzupassen und zu nutzen.*
	Zerlegung	*Zerlegung ist die Fähigkeit, ein Problem oder ein großes Projekt in kleinere Aufgaben und Bestandteile zu zerlegen und zu unterteilen, um auf diese Weise einen komplexen Prozess analysieren und bearbeiten zu können.*

	Abstraktion	*Abstraktion bedeutet, Sachverhalte zu vereinfachen, indem das Wichtigste identifiziert wird und unnötige Details ausgelassen werden. Die Abstraktion erlaubt es, über Dinge in unterschiedlichem Detailgrad nachzudenken.*
	Algorithmisches Denken	*Algorithmisches Denken ist die Fähigkeit, eine Reihe von Anweisungen nacheinander abzuarbeiten. Die Anweisungen bestehen aus einer endlichen Anzahl an Handlungsschritten.*
	Iteration	*Iteration ist die Fähigkeit, sich wiederholende identische Handlungen zusammenzufassen und mehrfach als Block durchzuführen. Gleichbleibende Einzelschritte können durch Iteration zusammengefasst dargestellt werden.*
	Debugging	*Debugging ist der Prozess und die Kompetenz, Fehler aufzufinden und zu beheben. Fehler beheben bedeutet, neue Lösungswege auszuprobieren, wenn ein Prototyp nicht wie vorgesehen funktioniert. Auf diese Art und Weise richtet sich das eigene Denken lösungsorientiert aus.*
	Evaluation	*Evaluation ist die Kompetenz, einen Algorithmus auf seine Richtigkeit und Effizienz hin zu überprüfen. Bei der Evaluation kann die funktionale Richtigkeit geprüft werden (Wurden alle Möglichkeiten in Betracht gezogen?)oder die nutzungsfreundliche Richtigkeit (Ist die Anwendung für die Nutzer einfach zu handhaben?).*

Die SchülerInnen entwickeln und vertiefen diese acht Kompetenzen während der Anwendung des informatischen Denkens im Unterricht. Dabei durchlaufen die Lernenden mit der Lehrperson vier aufeinanderfolgende Etappen, die im nächsten Kapitel vorgestellt werden.

Umsetzung des Computational Thinking im Unterricht/

Das Experimentieren im Sachunterricht oder das Lösen von Textaufgaben im Mathematikunterricht sind Aktivitäten, die von den SchülerInnen problemlösendes Denken erfordern und das Trainieren der acht grundlegenden Kompetenzen des Computational Thinking ermöglichen.

Beim Computational Thinking steht die Entwicklung einer Problemlösungsstrategie im Vordergrund. Die SchülerInnen erlernen das Aufstellen einer Handlungsabfolge, die zu einer Lösung der Aufgabe führt. Diese Handlungsabfolge, oder Algorithmus, kann anschließend von einem Menschen oder einem Computer durchgeführt werden.

Dieser Problemlöse-Prozess besteht aus vier aufeinander folgenden Etappen, die auf den acht grundlegenden Kompetenzen des Computational Thinking basieren.

Die vier Phasen einer Computational-Thinking-Aktivität (nach Reuter, 2020) in eigener Darstellung

1. Etappe: Problemdefinition

Die Problemstellung wird untersucht und mit anderen, bereits bearbeiteten, Aufgaben verglichen.

Kernfragen: Gibt es Übereinstimmungen oder gemeinsame Merkmale? Sind gemeinsame Muster zu anderen Aufgaben erkennbar?

Kompetenzen: Mustererkennung, Verallgemeinerung

2. Etappe: Planung

Das Problem wird in mehrere Einzelschritte zerlegt und komplexe Aufgaben werden vereinfacht. Gleichzeitig werden unwichtige Details weggelassen. Es gilt, den Kern des Problems zu erkennen und diesen kleinschrittig zu definieren.

Kernfragen: Welche Details können weggelassen werden? Was ist der Kern des Problems? Wie kann die Aufgabe in Teilschritte zerlegt werden?

Kompetenzen: Zerlegung, Abstraktion

3. Etappe: Umsetzung

Die einzelnen Handlungsschritte werden festgelegt. Die nutzende Person überlegt, wie er oder sie die Handlungen definieren und notieren kann, damit ein anderer diese ausführen kann.

Die Handlungsanweisungen (Algorithmen) werden präzise und detailliert und in einer bestimmten Reihenfolge vorgegeben. Wiederholungen der Handlungsanweisungen lassen sich in einem Block zusammenzufassen, mit dem Ziel, diese übersichtlicher zu gestalten und den Algorithmus zu optimieren.

Kernfragen: Wie können die einzelnen Handlungsanweisungen notiert werden? Gibt es sich wiederholende Schritte, die zusammengefasst werden können?

Kompetenzen: Algorithmisches Denken, Wiederholung der Handlungsanweisungen (Iteration)

4. Etappe: Überprüfung

Die Handlungsschritte werden umgesetzt: sie können zum vorgegebenen Ziel führen oder auch nicht. Im letzteren Fall muss der Algorithmus überarbeitet, Fehler gefunden und korrigiert werden (Debugging). Die vorher aufgestellten Schritte werden evaluiert.

Kernfragen: Wurde die Aufgabe gelöst, kann über mögliche Optimierungen nachgedacht werden. Gibt es einfachere, schnellere, effizientere Lösungen?

Kompetenzen: Debugging, Evaluation

Das bewusste Durchschreiten dieser Etappen unterstützt die SchülerInnen bei anderen schulischen Aktivitäten, wie zum Beispiel bei naturwissenschaftlichen Experimenten oder beim Lösen von Textaufgaben in der Mathematik.

Neben den acht Computational-Thinking-Kompetenzen spielen auch die 4K-Fertigkeiten eine grundlegende Rolle: Kommunikation, Kollaboration, Kreativität und Kritisches Denken. „Wenn Wissen nur passiv erworben wurde, ohne dass Skills daran beteiligt sind, wird es nur oberflächlich erlernt." *(Fadel et al., 2017, S. 124)*. Die 4K-Fertigkeiten sind transversale Kompetenzen, die unerlässlich sind, um die Kreativität, den Peer-Austausch bei der Zusammenarbeit und das kritische Auseinandersetzen mit den Technologien zu fördern.

Vor diesem Hintergrund benötigen die Kinder einen Lernkontext, in dem aktive Partizipation, Kreativität und Zusammenarbeiten mit Gleichaltrigen möglich ist. Die Förderung der Kreativität kann durch das Lernmotto: Projekt, Leidenschaft, Peers und Spiel[1] unterstützt werden.

Durch selbständiges Arbeiten in Projekten konstruieren die SchülerInnen ihr Wissen gemäß ihres Entwicklungsstandes. Die Projektarbeit lässt den SchülerInnen genügend Freiraum, Inhalte zu vertiefen und auf andere schulische Fächer zu erweitern.

[1] *Im Original: project, passion, peers, play (vgl. Resnick, 2017)*

Die Leidenschaft am Projekt führt dazu, dass die SchülerInnen ihre Arbeit pflegen, sich darum kümmern und sie mit ihrer Erfahrungswelt verknüpfen. Kreativität ist ein sozialer Prozess, dementsprechend ist die Komponente Peer von großer Bedeutung. Durch die Komponente Spiel werden die SchülerInnen ermutigt, Risiken einzugehen und Neues auszuprobieren. Fehler machen erhält in diesem spielerischen Umfeld einen neuen Stellenwert.

Mit Blick auf einen partizipativen und kommunikationsfördernden Lernkontext fällt der Lehrperson eine besondere Rolle zu. Für selbstständig arbeitende SchülerInnen gestaltet die Lehrperson eine Lernumgebung, in der sie sich nach eigenem Rhythmus entfalten können. Die Lehrperson steht für Feedback, Verständnisfragen und richtungsweisende Impulse zur Verfügung. Für Lernende hingegen, die mehr Anleitung und eine engere Begleitung benötigen, bietet die Lehrperson die nötige Hilfestellung an.

Dieses *"Scaffolding"* ist eine schrittweise Prozedur mit unterstützenden Strategien in der initialen Lernphase. Sobald der Lernende die Fähigkeit besitzt, eine bestimmte Teilaufgabe selbstständig zu bearbeiten, wird das Gerüst schrittweise wieder entfernt.

Auf der Website educoding.lu illustriert der Film über Computational Thinking an Grundschulen den Einsatz des informatischen Denkens in den Zyklen 1 bis 4.

www.educoding.lu

Inhalt des Starter-Kits/

Das Erlernen der **Computational Thinking** Kompetenzen, der mathematischen Kompetenzen und des Coding ist mit einer großen Vielfalt an analogem und digitalem Material möglich. Für die Lernaktivitäten in den ersten beiden Umsetzungsjahren wurde das Material nach dem Prinzip *"low floor, heigh ceiling" (Papert, 1980)* ausgesucht. Das bedeutet, dass das Material einfach zu handhaben ist und einen schnellen Einstieg ermöglicht. Gleichzeitig lässt es genügend Freiraum für komplexere Anwendungen und Kreativität.

Zusätzlich zu den bei Begleitheften ***"Logesch! oder?"*** und "EInfach Kodéieren" finden Sie im Starter-Kit folgendes Material:

- **6 Ozobots** *(mit Farbstiften, ActivityPack und 3 Würfeln)*

- **4 Kniwwelinos** *(mit Kabel, LED's und Sensoren)*

- **Lux-Robo Modikit** *(mit Zusatzmaterial)*

Wie Sie das Material im Unterricht einsetzen können, zeigen Ihnen die Unterrichtsideen in diesem Begleitheft.

Die begleitende Website www.educoding.lu hilft Ihnen bei der Auswahl von weiteren Materialien. Rezensionen zu verschiedener Hard- und Software stehen dort zur Orientierung zur Verfügung.

www.educoding.lu

Erklärungen zu den Unterrichts-ideen/

Die Unterrichtsideen in diesem Begleitheft ermöglichen, jede Schulwoche mindestens eine Aktivität im Bereich des Computational-Thinking-durchzuführen.

Die Unterrichtsideen sind wie folgt aufgebaut:

1. *Unplugged-Aktivitäten: Einstieg in das informatische Denken anhand von Aktivitäten, die kein digitales Material erfordern. Die SchülerInnen lernen auf spielerische Art und Weise informatisches Denken. Das Begleitheft "Logesch! oder?" mit 12 Unterrichtsideen ist auf der Webseite von educoding erhältlich.*

Begleitheft *"Logesch! oder?"* mit 12 Unterrichtsideen zu Unplugged-Aktivitäten

Vertiefende Informationen auf: https://csunplugged.org/en/what-is-computer-science/.

2. *Aktivitäten mit Ozobots: Diese Roboter ermöglichen einen spielerischen Einstieg ins Programmieren mit Hilfe von Farbcodes. Nähere Informationen auf: https://ozobot-deutschland.de/.*
3. *Blockbasierte Programmiersprache mithilfe von Ozobots: Die Ozobots werden am Computer oder Tablet anhand einer blockbasierten Programmiersprache programmiert. Nähere Informationen auf: https://ozoblockly.com/.*
4. *Einsatz von Scratch: Ist die blockbasierte Sprache bekannt, können kleine Projekte in Scratch leicht umgesetzt werden. Nähere Informationen auf: https://scratch.mit.edu/.*
5. *Umgang mit Kniwwelino: Diese Platine kann mit Scratch programmiert und für kreative Aktivitäten und Projekte genutzt werden. Nähere Informationen auf: https://www.kniwwelino.lu/.*
6. *Einsatz von ModiKit: Anhand der Module dieser Hardware lernen die SchülerInnen eigene Projekte aufzubauen. Vertiefende Informationen auf: https://de.luxrobo.co.uk/.*

Dieses Konzept baut anhand der Unterrichtsideen das informatische Denken von *„Unplugged"*-Aktivitäten über Programmieren mit Farbcodes bis hin zum Kodieren von Platinen und Modulen auf. Jede Unterrichtsidee geht, unabhängig von der Aktivität und dem Thema, auf folgende Aspekte ein:

- Die Vernetzung der Aktivität mit Kompetenzen des Lehrplans, zunächst vorrangig aus dem Fach Mathematik.
- Die Vernetzung der Aktivität mit dem Alltag der Lernenden und ihrer *(digitalen)* Umwelt.
- Die Vernetzung der Aktivität mit Konzepten aus dem Bereich der Informatik.

Die Unterrichtsideen sind zeitlich auf eine Unterrichtseinheit ausgelegt. Ihre Vernetzung mit verschiedenen Fächern und/oder Themen erlaubt eine zeitlich umfassendere und transversale Bearbeitung und Umsetzung.

Am Ende dieses Heftes finden Sie Weblinks zu weiteren Ideen.

Glossar/

21st Century Skills
Eine Skill *(englisch)* ist eine Kompetenz. Die Kompetenzen des 21. Jahrhunderts definieren die Kompetenzen, die den Lernenden ermöglichen, sich in dieser Welt zurechtzufinden und zu aktiven BürgerInnen zu werden.

4K-Skills
Die 4K-Skills *(4K-Fertigkeiten)* stehen für vier Kompetenzen, die im Zeitalter der Automatisierung wichtiger sind denn je: Kommunikation, Kollaboration, Kreativität und Kritisches Denken.

Abstraktion
Weglassen von Einzelheiten und unwichtigen Details.

Algebra
Teilgebiet der Mathematik, das sich mit Rechenoperationen und deren Eigenschaften befasst.

Algorithmus
Eine endliche Auflistung mehrerer Aktionen in einer präzisen Reihenfolge.

Arithmetik
Teilgebiet der Mathematik, das sich mit dem Rechnen befasst.

Bildung 4.0
Das Zusammenführen von Medienbildung und Informatik führt zu einer erweiterten Medienbildung, auch Bildung 4.0 genannt. Diese beinhaltet die Mediennutzungs-Kompetenz, die Medien-Kritik und die technologische Perspektive.

Binär
Bestehend aus zwei Zuständen. Computer kennen nur zwei Zustände: entweder fließt Strom *(Zustand 1)* oder es fließt kein Strom *(Zustand 0)*.

Bug
Ein Bug *(englisch)* ist ein Käfer. In der Informatik wird ein Fehler in einem Programm als Bug bezeichnet.

Coding
Kodieren oder Programmieren heißt, das Aufschreiben einer Handlungsanweisung, eines Algorithmus, in einer für eine Maschine lesbare Art und Weise.

Computational Thinking
Informatisches oder problemlösendes Denken.

Computer Science
Siehe Informatik.

Daten
Zeichen oder Symbole, die Informationen darstellen.

Datenanalyse
Untersuchung von Daten zwecks Weiterverarbeitung.

Datengewinnung
Sammeln von Daten für einen bestimmten Zweck. Daten können manuell oder automatisch *(durch Algorithmen, Sensoren)* erhoben werden.

Debugging
Fehlerbehebung *(siehe Bug)*.

Digital
In binärer *(siehe binär)* Form, also bestehend aus zwei Zuständen. Diese Form ist für Computer oder digitale Maschinen lesbar und verarbeitbar.

Digital Literacy
Digitale Kompetenz, neben Lesen, Schreiben und Rechnen ist die *Digital Literacy* heute die vierte Kulturtechnik, die an Schulen gelehrt wird.

Digital Native
Eine Person, die in der digitalen Welt aufgewachsen ist.

Evaluation
Untersuchung und Bewertung einer Aufgabe, eines Prozesses.

Geometrie
Teilgebiet der Mathematik, das sich mit Punkten, Geraden, Winkeln, Flächen und Körpern befasst.

Hardware
Ein Oberbegriff für die elektronischen und mechanischen Bestandteile eines digitalen Gerätes.

Informatisches Denken
Siehe Computational Thinking.

Informatik
Das Wort Informatik setzt sich zusammen aus *"Information"* und *"automatisch"*. Die Informatik behandelt das *(überwiegend)* automatische Verarbeiten von Informationen.

Iteration
Mehrfaches Wiederholen.

Kodieren
Siehe *Coding*.

Loop
Eine Schleife, eine Sequenz aus Kommandos, die mehrfach ausgeführt werden.

Medienbildung
Die Medienbildung lehrt einen verantwortungsbewussten und sicheren Umgang mit den Medien und weist auf mögliche Gefahren bei der Anwendung dieser hin. Sie hat zum Ziel, die SchülerInnen zu aktiven und reflektierenden BürgerInnen im Umgang mit digitalen Medien zu erziehen.

Medienkompetenz
Die Fähigkeit, Medien und ihre Inhalte den eigenen Zielen und Bedürfnissen entsprechend kompetent zu nutzen.

Medienkritik
Die Medienkritik zielt auf ein reflektierendes Umgehen mit Informationen. Problematische gesellschaftliche Prozesse werden analytisch erfasst und kritisch hinterfragt. Die Menschen wenden zudem ihr Wissen reflexiv auf sich selbst und auf ihr Handeln an und berücksichtigen die sozialen Konsequenzen der Medienentwicklung.

Mediennutzung
Die Mediennutzung umfasst das Anwenden oder die Nutzung von Medien und das aktive Erstellen von Medieninhalten.

Modellieren
Erstellen eines vereinfachten Abbildes der Wirklichkeit.

Mustererkennung
Bei mehreren Objekten Regelmäßigkeiten, Wiederholungen, Ähnlichkeiten oder Gemeinsamkeiten erkennen.

Pattern
Englisch: ein Muster, eine gleichförmige Wiederholung.

Problem solving
Problemlösen.

Programmieren
Siehe *Coding*.

Programmcode
Niederschrift eines Algorithmus, der festlegt, in welcher Reihenfolge Befehle durchgeführt werden.

Robotik
Wissenschaftliche Disziplin, die sich mit der Entwicklung von Robotern beschäftigt.

Scaffolding
Schrittweise Prozedur mit unterstützenden Strategien in der initialen Lernphase.

Scratch
Blockbasierte Programmiersprache für Kinder. Scratch wurde am MIT *(Massachusetts Institute of Technology)* entwickelt.

Simulation
Eine Simulation ist das Durchführen von Experimenten an einem Modell, also nicht am realen System.

Skills
Fertigkeiten, Kompetenzen.

Statistik
Die Zusammenfassung bestimmter Methoden zur Analyse von erhobenen Daten.

Software
Sammelbegriff für Programme, Apps und ihre zugehörigen Daten.

Unplugged
Ohne Strom.

Verallgemeinerung
Generalisierung, das Bündeln von gemeinsamen Eigenschaften in eine Einheit *(Konzept der Modellierung)*.

Virtuelle Welt
Eine Welt, die nicht real existiert und nur über einen Computer oder ein anderes digitales Gerät betreten werden kann.

Wahrscheinlichkeit
Probabilität.

Zahlenbegriff
Konkrete Vorstellung der Mengen, die durch Zahlen dargestellt werden.

Zerlegung
Aufteilung in mehrere kleinere Teile/Schritte.

Ozobot Farbcodes

Ozobot Farbcode-Übersicht/

Geschwindigkeit

langsam

normal

schnell

Turbo

Schnecke für 3 Sekunden

Turbo für 3 Sekunden

Richtung

nach links

nach rechts

geradeaus

umkehren *(halbe Drehung)*

nach links springen

nach rechts springen

geradeaus springen

umkehren am Linien-Ende
(halbe Drehung)

Timer

warten für 3 Sekunden

Timer starten *(30 Sekunden)*

Timer stoppen

Zähler

setze Punktezähler auf 5

setze Kreuzungszähler auf 5

setze Abbiegungszähler auf 5

setze Farbwechselzähler auf 5

Zähler +1

Zähler -1

Spielstand *(am Linien-Ende)*

Gewonnen!

Verloren!

Tanzbewegungen

Tornado

zickzack fahren

ganze Drehung

rückwärts fahren

Flächeninhalte und Umfänge/

Material

- Schwarzer Stift
- Weiße Blätter

Bezug zum Lehrplan

Mathématiques : Espace et formes : Combiner des connaissances géométriques et arithmétiques afin de résoudre des problèmes mathématiques

Bezug zum Lebensalltag

Geometrische Formen mit gleichem Flächeninhalt können unterschiedliche Umfänge haben. In dieser Unterrichtseinheit wird dies praktisch mit dem Ozobot veranschaulicht.

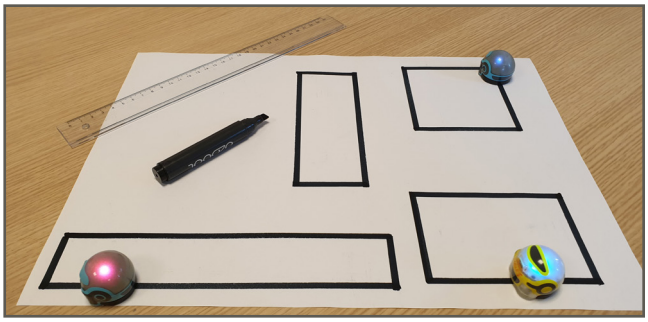
Ablauf

Ein Rechteck hat einen Flächeninhalt von 400 cm^2. Zeichne mehrere verschiedene Rechtecke, welche den gleichen Flächeninhalt aber verschiedene Umfänge haben.

Setze den Ozobot an eine Ecke der Rechtecke und lasse ihn den Umfang abfahren. Stoppe die Zeit, die er dazu braucht.

Vergleiche die Zeiten. Wie viel Zeit braucht er bei den verschiedenen Rechtecken?

Bei welchem Rechteck braucht er am wenigsten Zeit, bei welchem am meisten?



Vertiefung:

Die SchülerInnen bestimmen einen anderen Wert des Flächeninhaltes und stellen ihrem/ihrer MitschülerIn die gleiche Aufgabe.

Tipps und Tricks:

Bei Rechtecken mit größeren Flächeninhalten fallen die Ungenauigkeiten beim Zeitmessen weniger auf.

Challenge Cards/

Material

- Ozobots
- Stifte
- Arbeitsblatt mit den Farbcodes
- Challenge-Karten

Bezug zum Lehrplan

Mathématiques : Espace et formes : Analyser et représenter des figures géométriques dans le plan et l'espace

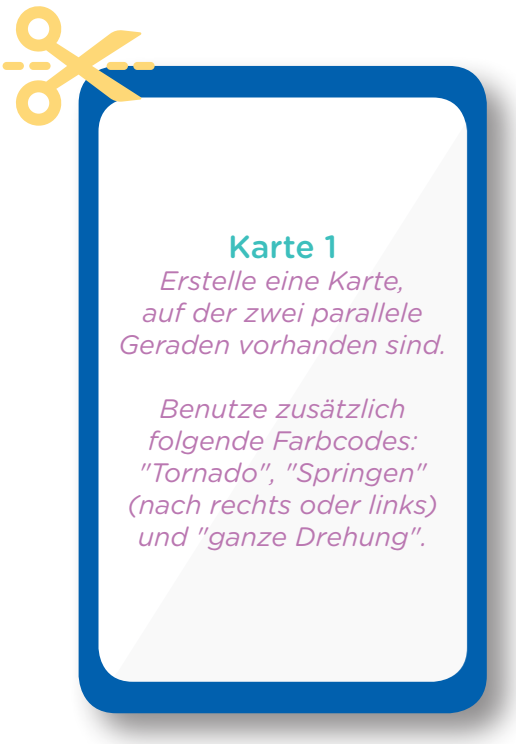
Bezug zum Lebensalltag

Die SchülerInnen lernen die Farbcodes kennen. Gleichzeitig wiederholen sie geometrische Begriffe.

Ablauf

Die SchülerInnen arbeiten in Kleingruppen an den verschiedenen Karten.

Sie zeichnen mit dem Stift die Angaben der Karten auf ein Blatt und lassen diese Zeichnung vom Ozobot abfahren.



Karte 1

Erstelle eine Karte, auf der zwei parallele Geraden vorhanden sind.

Benutze zusätzlich folgende Farbcodes: "Tornado", "Springen" (nach rechts oder links) und "ganze Drehung".

Karte 2

Erstelle einen Pfad, der ein Rechteck mit einem Flächeninhalt von 40 cm^2 darstellt.

Benutze zusätzlich folgende Farbcodes: "Turbo" und "ganze Drehung".

Karte 3

Erstelle einen Pfad, der ein Dreieck mit einem stumpfen Winkel darstellt.

Benutze zusätzlich folgende Farbcodes: "Umkehren" und "Warten für 3 Sekunden".

Karte 4

Erstelle einen Pfad, der ein Rechteck mit einem Umfang von 40 cm darstellt.

Benutze zusätzlich folgende Farbcodes: "Schnecke für 3 Sekunden" und "Schnell".

Karte 5

Zeichne zwei Geraden. Diese Geraden sollen sich in einem spitzen Winkel treffen.

Benutze zusätzlich folgende Farbcodes: "Umkehren" und "Tornado".

Karte 6

Erstelle einen Pfad, der wenigstens zwei rechte Winkel hat.
Benutze zusätzlich folgende Farbcodes: "Warten für 3 Sekunden" und "Tornado".

Karte 7

Erstelle einen Pfad, der einen Kreis mit einem Radius von 6 cm darstellt.

Benutze zusätzlich folgende Farbcodes: "Warten für 3 Sekunden" und "Ganze Drehung".

Karte 8

Erstelle einen Pfad, der eine Strecke mit den Endpunkten A und B darstellt.

Benutze zusätzlich folgenden Farbcode: "Zickzack fahren".

Karte 9

Erstelle einen Pfad, der ein gleichseitiges Dreieck darstellt.

Benutze zusätzlich folgenden Farbcode: "Tornado".

Karte 10

Erstelle einen Pfad, der ein Parallelogramm darstellt.

Benutze zusätzlich folgenden Farbcode: "Rückwärts fahren".

Karte 11

Erstelle einen Pfad, der wenigstens zwei Winkel enthält. Ein Winkel muss einen rechten Winkel darstellen.

Benutze zusätzlich folgende Farbcodes: "Ganze Drehung" und "Warten für 3 Sekunden".

Vertiefung

Karte 12

Erstelle einen Pfad, der ein Trapez darstellt.

Benutze zusätzlich folgenden Farbcode: "Turbo für 3 Sekunden".

Vertiefung

Karte 13

Erstelle einen Pfad, der eine Raute darstellt.

Benutze zusätzlich folgenden Farbcode: "Schnecke für 3 Sekunden".

Vertiefung

Karte 14

Erstelle einen Pfad, der ein Rechteck mit einem Umfang von 400 mm darstellt.

Benutze zusätzlich folgende Farbcodes: "Zickzack fahren" und "ganze Drehung".

Vertiefung

Karte 15

Zeichne zwei Geraden, die einen überstumpfen Winkel bilden.

Benutze zusätzlich folgende Farbcodes: "Ganze Drehung" und "Tornado".

Vertiefung

Karte 16

Erstelle einen Pfad, der ein Dreieck mit einem Flächeninhalt von 30 cm^2 darstellt.

Benutze zusätzlich folgende Farbcodes: "Schnell" und "Warten für 3 Sekunden".

Vertiefung

Karte 17

Erstelle einen Pfad, der einen stumpfen Winkel, einen rechten Winkel und einen überstumpfen Winkel hat.

Benutze zusätzlich folgende Farbcodes: "Tornado", "ganze Drehung" und "Warten für 3 Sekunden".

Vertiefung:

Welche der Ozobot-Farbcodes sind palindrom? Wie viele Farbcodes - bestehend aus drei, vier, fünf... Blöcken - können mit den vier Ozobot-Farben schwarz, rot, blau und grün erstellt werden?

Diese Erkenntnisse können auf das Bilden von Zahlen - bestehend aus drei, vier, fünf... Ziffern - übertragen werden.

Tipps und Tricks:

Die SchülerInnen erstellen ihre eigenen Challenge-Karten. Dabei setzen sie sich kreaktiv mit dem Ozobot und den Farbcodes auseinander.

Ozobot: Kopfrechnen/

Material

- Ozobots
- Farbstifte
- Arbeitsblätter

Bezug zum Lehrplan

Mathématiques : Nombres et opérations : Reconnaître des structures et des règles arithmétiques

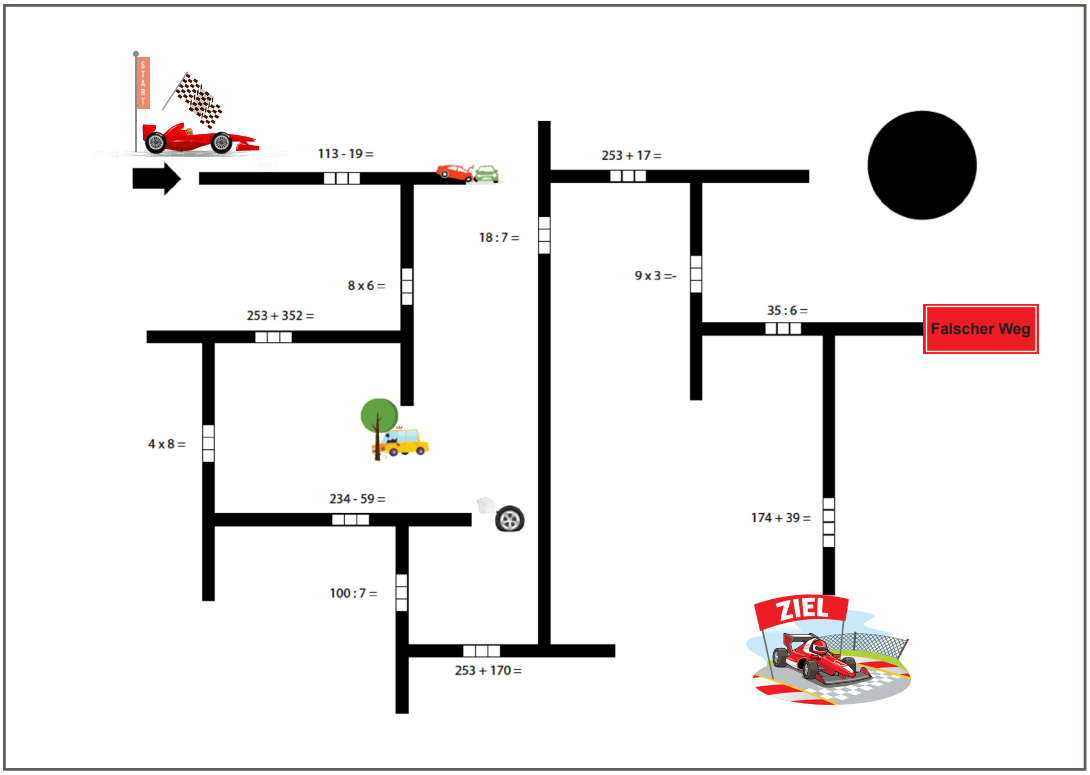
Bezug zum Lebensalltag

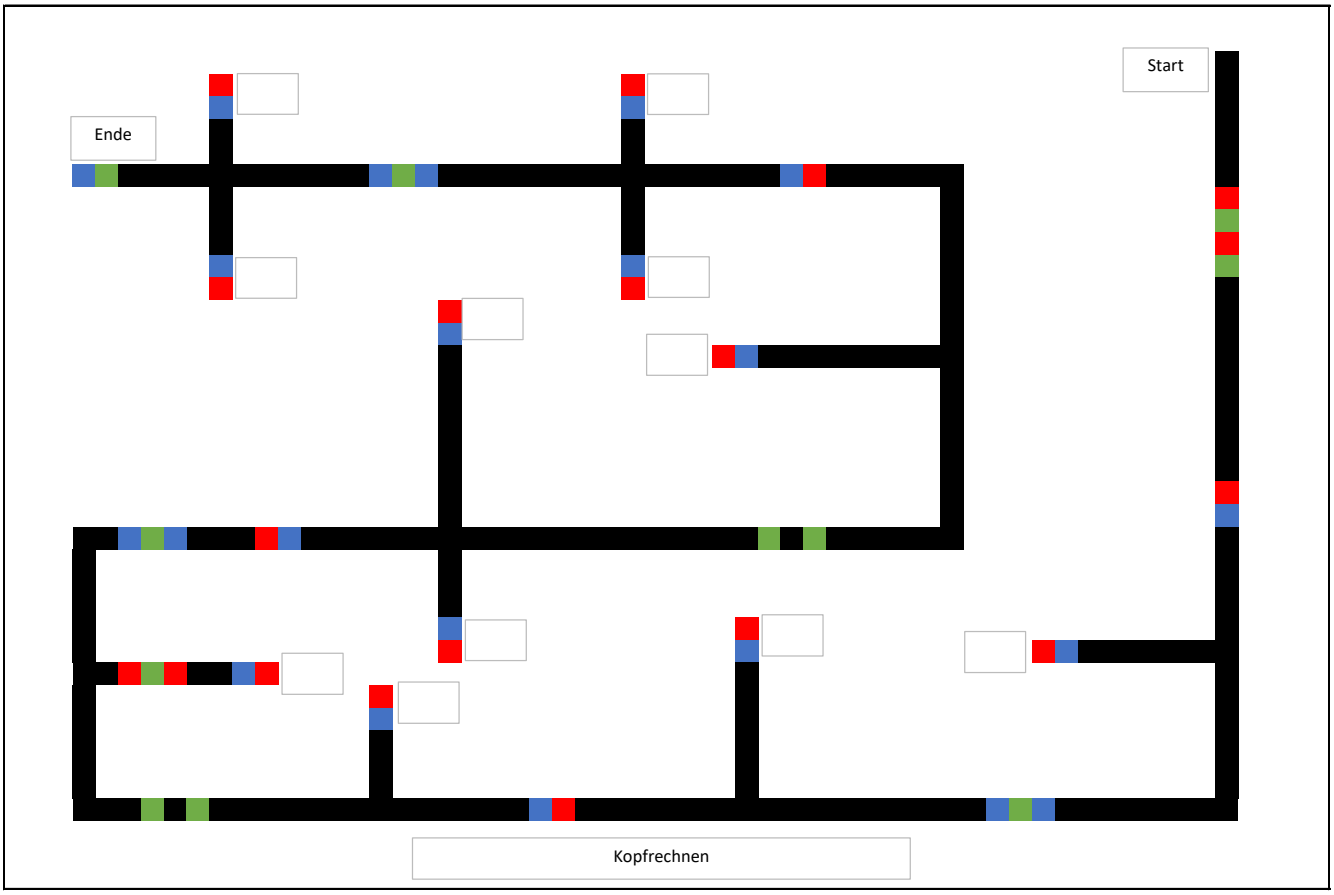
Mit dieser Unterrichtseinheit üben die SchülerInnen ihre Rechenfähigkeiten und lernen zudem, den Ozobot mit Hilfe von Farbcodes richtig zu programmieren.

Ablauf

Aktivität 1:

Rennfahrerin Rosi steht auf der Rennbahn und möchte dieses letzte Rennen auch noch gewinnen. Leider war sie gestern verunglückt und hatte eine kleine Gehirnerschütterung. Nun weiß sie nicht mehr, wie sie diese Rechnungen schnellstmöglich im Kopf rechnen kann. Kannst du ihr helfen, so schnell wie möglich ans Ziel zu gelangen, ohne von der Strecke abzukommen?





Aktivität 2:

Ein spannendes Rennen gegen den Roboter! Starte den Ozobot und addiere alle Zahlen, an denen der Roboter vorbeifährt. Schaffst du es richtig zu rechnen, bevor der Roboter am Ziel ankommt?

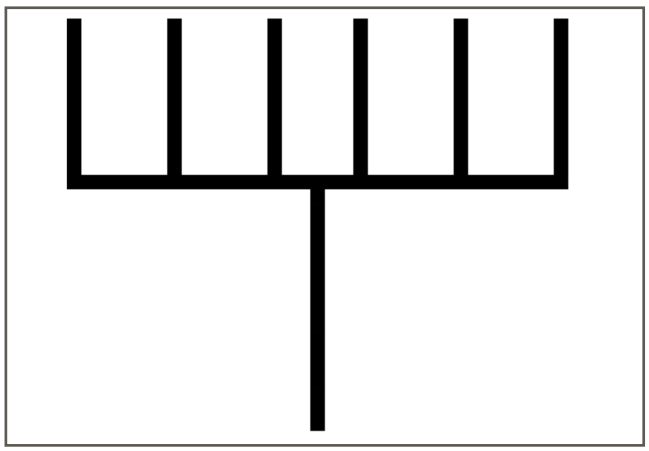
Die SchülerInnen setzen ihren Ozobot am Startpunkt auf. Der Roboter sucht sich einen Weg durch das Labyrinth. Immer wenn er an einer Zahl vorbeikommt, sollen die SchülerInnen die Zahlen addieren, subtrahieren oder multiplizieren.

Variationen: Multiplikationen einbauen.
Mit der Vorlage *(siehe QR-Code)* können ganz einfach in Excel neue Aufgabenkarten gezeichnet werden.

Vorlage für weitere Aufgabenkarten
(Excel-Datei)

Aktivität 3:

Die SchülerInnen setzen ihren Ozobot unten am Startpunkt auf. Der Ozobot sucht sich nach dem Zufallsprinzip einen Arm aus. Über jedem Arm steht eine Rechnung. Diese Rechnung wird im Kopf gerechnet.



Vertiefung:

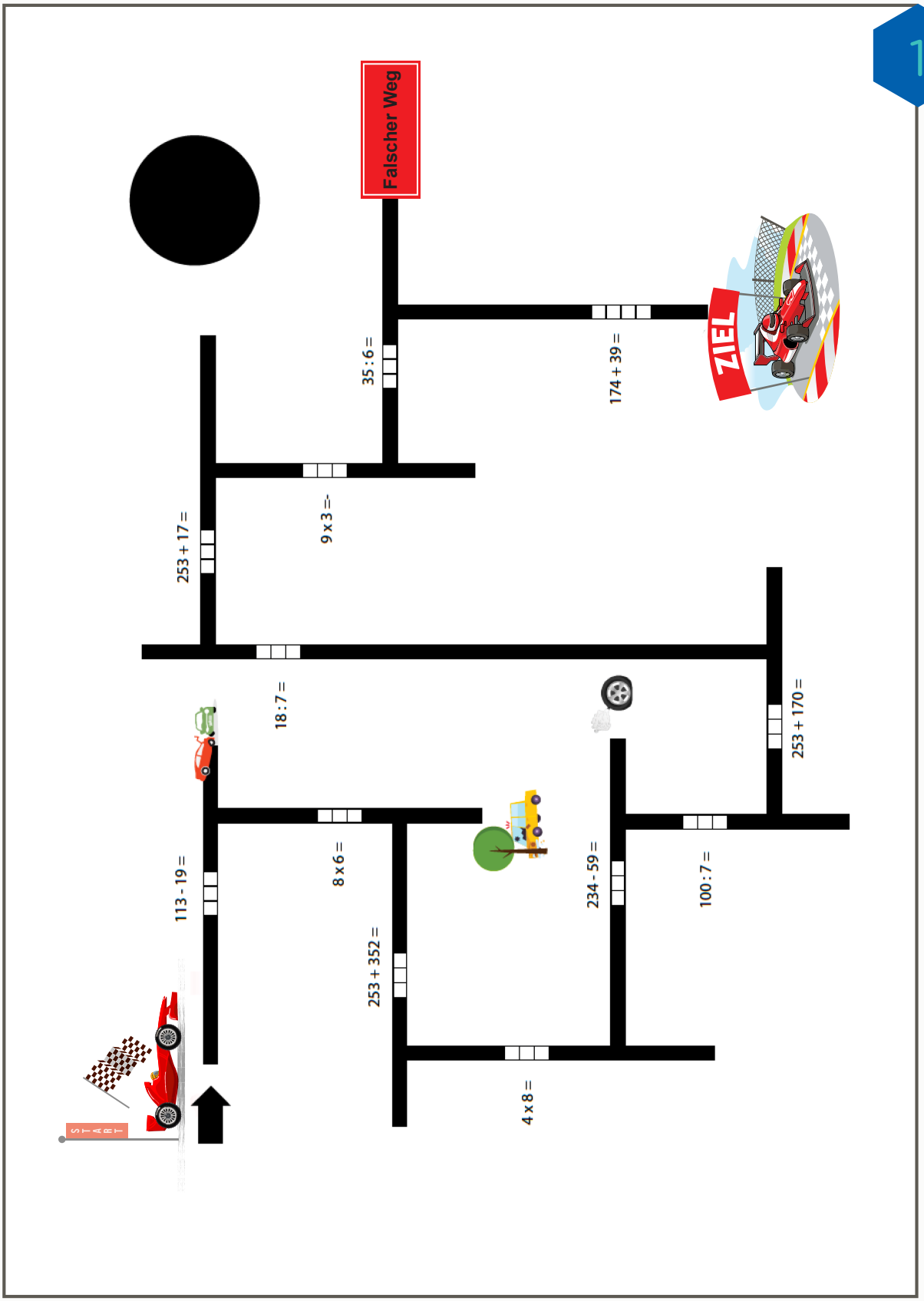
Die SchülerInnen können selbst die Aufgaben erstellen und ihrem/ihrer MitschülerIn die Aufgabe zum Ausrechnen geben.

Tipps und Tricks:

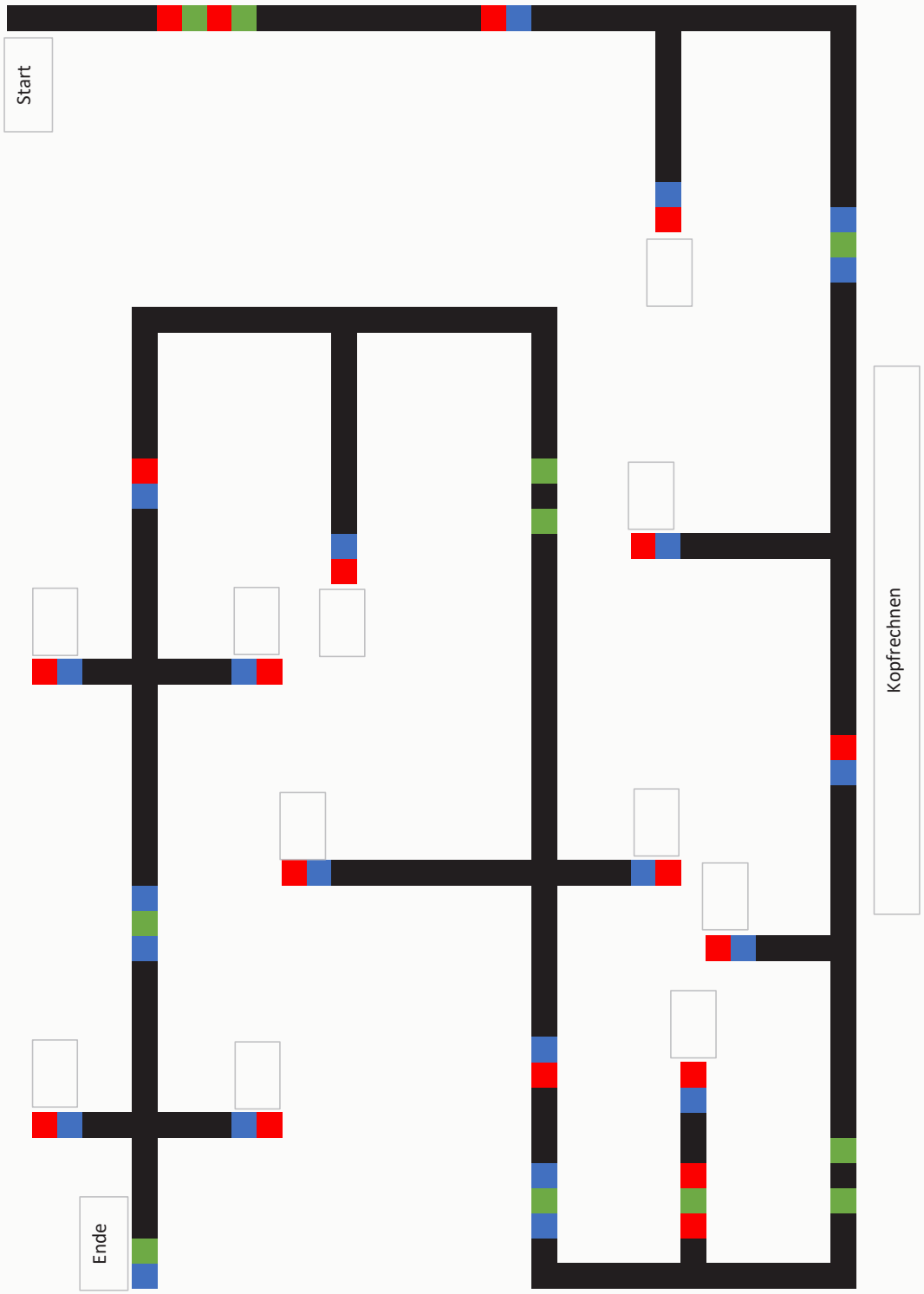
Diese Aufgaben lassen sich unterschiedlich differenzieren. Der Zahlenraum kann jeweils an die Kompetenzen der SchülerInnen angepasst werden *(wie z.B. Rechnen mit Dezimalzahlen, Brüchen...)*.

Kopieren Sie die Aufgabenkarten auf das A3-Format und zeichnen Sie gegebenenfalls die Linien mit dem Ozobot-Stift nach.

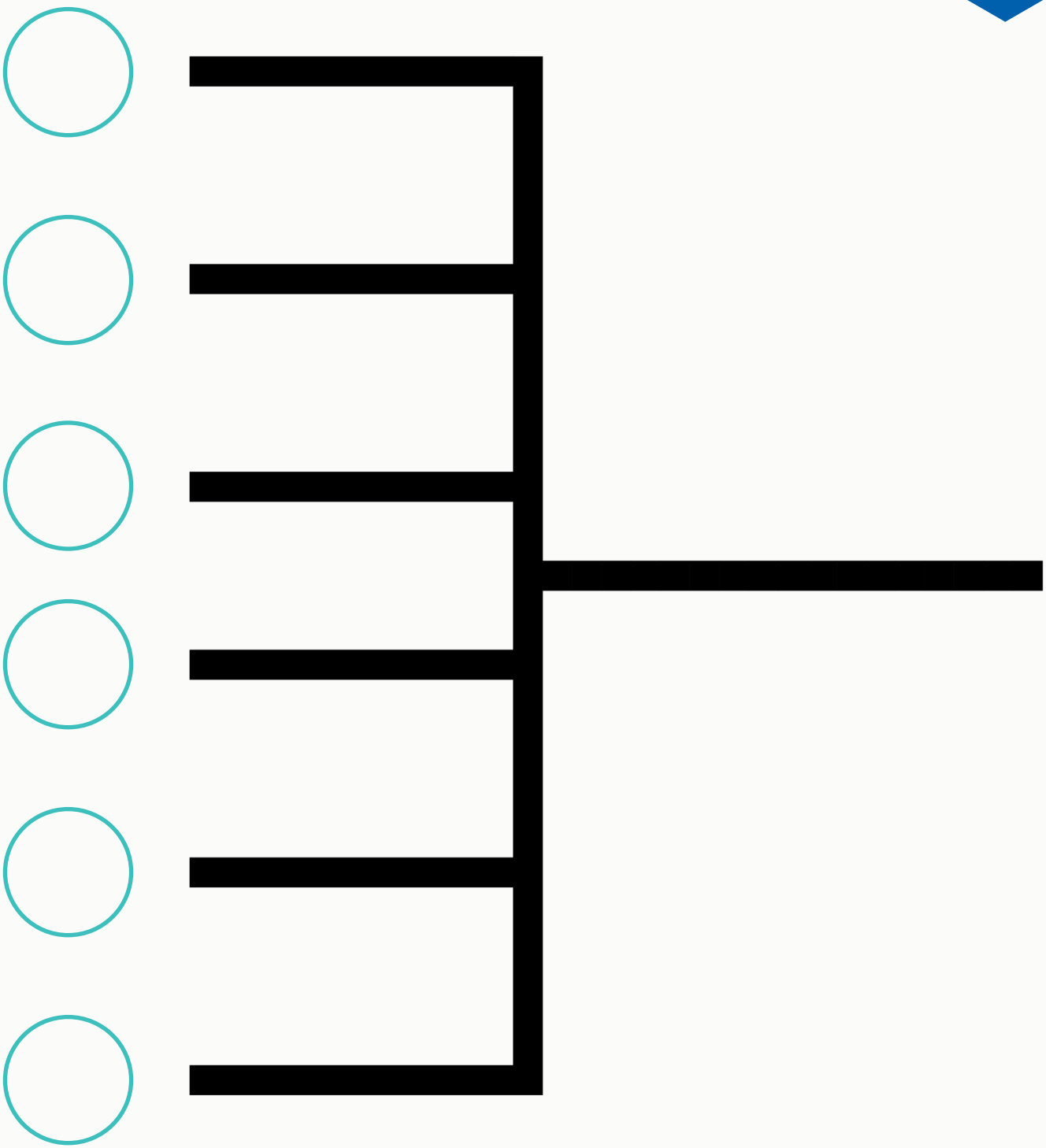
1


Falscher Weg
ZIEL
35 : 6 =
174 + 39 =
9 x 3 =
253 + 17 =
18 : 7 =
253 + 170 =
113 - 19 =
8 x 6 =
234 - 59 =
100 : 7 =
253 + 352 =
4 x 8 =
START

2



3



Interaktive Geschichte

Material

- Ozobot

Bezug zum Lehrplan

Langues : Production écrite : Rédiger différents types de textes

Bezug zum Lebensalltag

Interaktive Geschichten, die keinen festen Verlauf haben, sind immer wieder interessant, da sie anders verlaufen. Es gibt auch Apps, Bücher, Filme und Hörspiele nach diesem Prinzip.

Ablauf

Bei dieser Aktivität schreiben die SchülerInnen eine interaktive Geschichte, bei der die Leser selbst auswählen, wer oder was die Hauptfigur der Geschichte ist und wie sich die Geschichte entwickeln wird. Dazu müssen die SchülerInnen folgenden Etappen berücksichtigen:

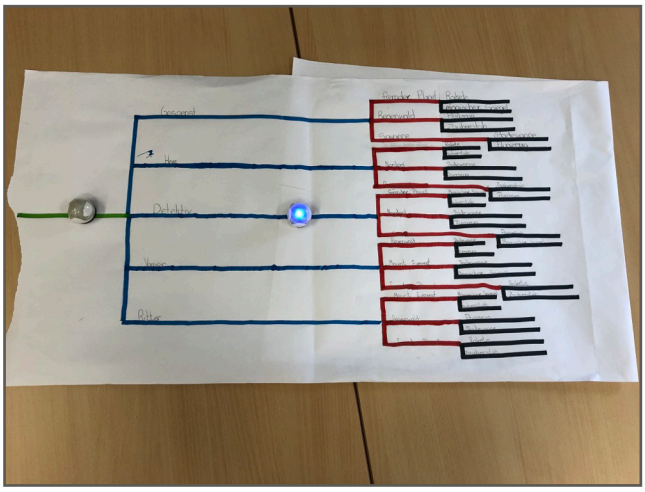
1. Hauptfiguren aussuchen und ausführlich beschreiben.

2. Verschiedene Orte aussuchen, an denen die Geschichte spielt.

3. Verschiedene Handlungsmöglichkeiten aussuchen. Beispiel: imaginäre Reisemittel zu unterschiedlichen Erdteilen definieren und dann kurz beschreiben, was die Hauptfigur an diesem fremden Ort erlebt.

4. Alle möglichen Kombinationen der Geschichte(n) auf einem Plan festhalten.

Es ist nicht möglich, alle Kombinationen auszuschöpfen und zu erzählen: hier kommt der Ozobot ins Spiel. Der Ozobot entscheidet zufällig an einer Kreuzung, welche Richtung er nimmt. Die SchülerInnen lassen den Ozobot starten und prägen sich den Weg, den er nimmt, ein. Mit diesen Elementen, die Ozobot ausgewählt hat, schreiben die SchülerInnen dann ihre Geschichte: also zum Beispiel: Ritter, Badewanne als Reisemittel, Reiseziel: Mount Everest, Begegnung mit Yeti.



Vertiefung:

Bei der Aufzeichnung des Plans der Geschichte(n) brauchen die SchülerInnen eventuell Hilfe, damit dieser tatsächlich für den Ozobot befahrbar ist. Achtung: der Plan sollte nur in eine Richtung befahrbar sein und nicht gewendet werden können. Mögliche Erweiterung: die Geschichte als interaktive Geschichte mittels Twine *(siehe https://twinery.org/)* verfassen, sodass sie online lesbar und spielbar ist.

Tipps und Tricks:

Diese Aufgabe kann im Sprachunterricht integriert werden.

Twinery:
Tool zum Erstellen interaktiver Geschichten

Würfel Challenge

Material

- Ozobot
- Ozobot-Würfel
- Stoppuhr

Bezug zum Lehrplan

Mathématiques : Grandeurs et mesures : Utiliser des grandeurs dans des situations de la vie courante

Bezug zum Lebensalltag

Die SchülerInnen erkennen den Zusammenhang zwischen den vom Ozobot abgefahrenen Strecken und der abgelaufenen Zeit.

Ablauf

Die SchülerInnen setzen sich in Gruppen zusammen. Jede Gruppe erhält drei Ozobot-Würfel und einen Ozobot. Zwei Würfel haben Farbcodes auf den Seiten abgebildet, ein Würfel Zeitangaben.

Jede Gruppe wirft einmal die drei Würfel. Die Zeitangabe gibt die Dauer in Sekunden an, die der Ozobot in Bewegung sein soll. Nach dieser Zeitspanne muss er stehen bleiben. Die anderen beiden Würfel zeigen die Farbcodes, die mindestens einmal benutzt werden müssen.

Die SchülerInnen zeichnen jetzt einen passenden Pfad. Es dürfen noch weitere Farbcodes benutzt werden. Die SchülerInnen testen immer wieder aus und stoppen dabei die Zeit.

Vertiefung:

Zusätzlich kann auch ein Spiel-Würfel benutzt werden. Er gibt die Anzahl der Kreuzungen an, die im Parcours vorkommen müssen.

Die SchülerInnen basteln ihre eigenen Challenge-Würfel mit ihren bevorzugten Farbcodes.

Tipps und Tricks:

Die Aktivität kann als Wettkampf durchgeführt werden: Gewonnen hat die Gruppe, die am genauesten an ihre Zeitangabe herankommt.

Die gewürfelten Codes können zu Beginn einmal aufgezeichnet und ausgeschnitten werden. So können sie überall auf dem Pfad hingelegt werden, und es kann getestet werden, wie sich der Ozobot-Roboter verhält!

Ozobot
Ozoblockly

Shapetracer/

Material

- Ozobot
- Tablet oder Computer
- Internetseite: https://games.ozoblockly.com/shapetracer-basic

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

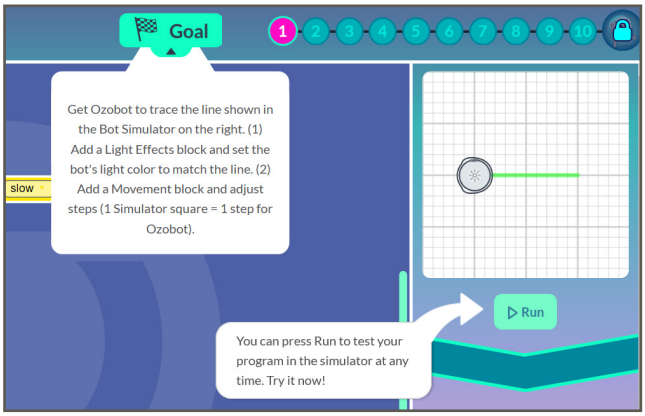
Bezug zum Lebensalltag

Die Applikation Shapetracer bietet die Möglichkeit, unterschiedliche Übungen mit dem Ozobot mit Hilfe der Blockly-Programmiersprache im Browser durchzuführen. Die Blockly-Programmiersprache wird später ebenfalls beim Programmieren in Scratch und Kniwwelino benutzt.

Ablauf

Rechts oben im Bild ist der Schwierigkeitsgrad hervorgehoben, den die SchülerInnen bearbeiten.

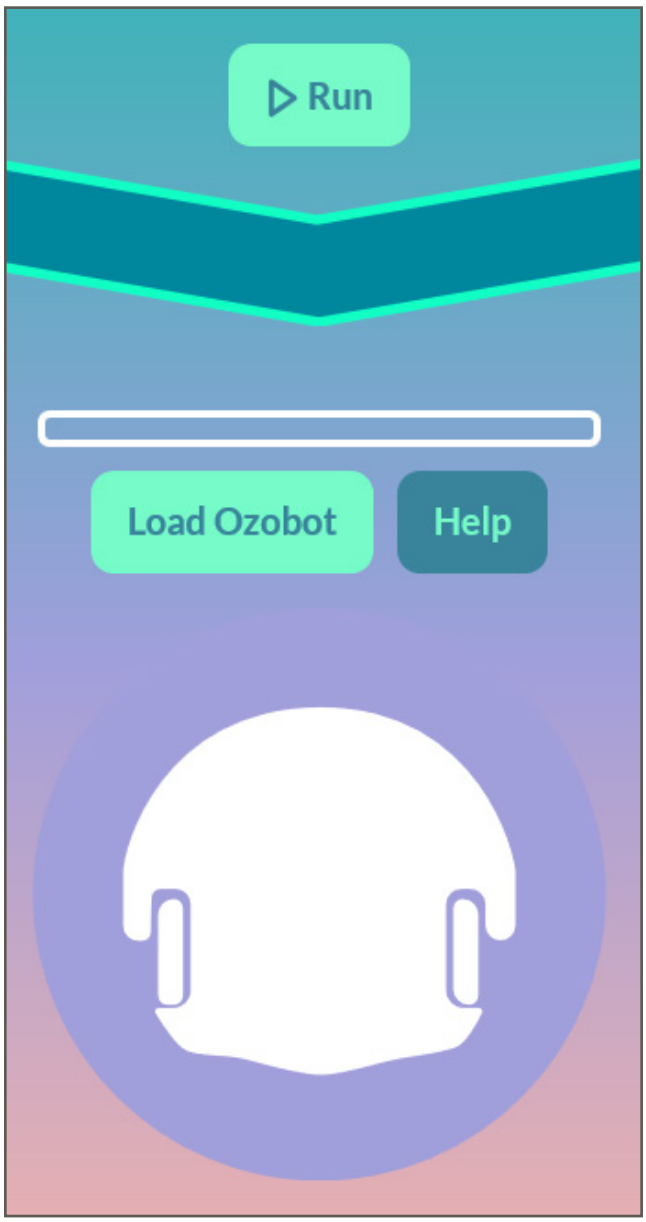
Die SchülerInnen ziehen die einzelnen Befehle von links "*Movement*" ins Programmierfeld in der Mitte. Sie können zum Beispiel den Begriff "*move forward distance 1 step speed slow*" nehmen.



In der Mitte des oberen Bereichs steht "*Goal*". Jeder Schwierigkeitsgrad hat ein dem Niveau angepasstes Ziel. Dieses Ziel ist im rechten Bild durch die farbigen Striche erkennbar. Die Programmierung kann zu jeder Zeit durch Klicken auf "*Run*" ausprobiert werden.

Haben die SchülerInnen die Aufgabe richtig erarbeitet, kommen sie automatisch einen Schwierigkeitsgrad weiter.

Die SchülerInnen können die Programmierung auch auf einen Ozobot hochladen, indem sie diesen auf die entsprechende Stelle auf der Seite hinlegen und auf *"Load Ozobot"* drücken.



Vertiefung:

Nachdem die SchülerInnen den jeweiligen Schwierigkeitsgrad bestanden haben, können sie sich gegenseitig ein neues Ziel geben, wie zum Beispiel: "Versuche, mit dem Ozobot ein Rechteck zu zeichnen."

Weitere Aufgaben gibt es unter:

https://games.ozoblockly.com/ozotown-basic

https://games.ozoblockly.com/shapetracer-advanced?lang=en

Tipps und Tricks:

Für diese Arbeit eignen sich Zweiergruppen, da die SchülerInnen sich gegenseitig verbessern und unterstützen können.

Labyrinth entwerfen

Material

- Ozobot
- Tablet oder Computer
- Karton
- Wasserflaschen,
- Eisstiel und Holzstäbchen
- Klebepistole
- Strohhalme
- Papier
- ...

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Bezug zum Lebensalltag

Diese etwas längere Unterrichtseinheit (Zeitaufwand 2-3 Stunden) zielt darauf ab, dass sich die SchülerInnen nicht nur auf einem vorgezeichneten Plan orientieren, sondern kreativ ihr eigenes Labyrinth erschaffen.

Das Planungsbüro Tobozo in Luxemburg ist damit beauftragt, die besten Labyrinthe für Luxemburg zu entwerfen. Dazu haben sie uns eine Planungsvorlage zugesendet.

Die SchülerInnen tragen auf dieser Vorlage zuerst ihr Labyrinth mit den genauen Koordinaten des Materials ein. Anschließend bauen sie ihr Labyrinth und programmieren den Ozobot so, dass er selbstständig durch das Labyrinth fährt.

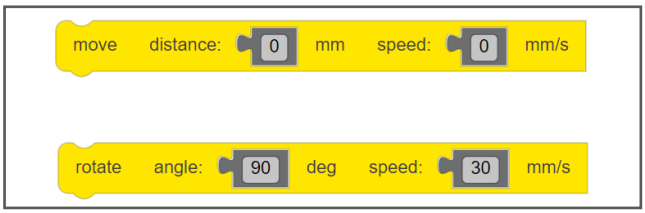
Ablauf

Zuerst erstellen die SchülerInnen ein Netz von 10 x 10 Quadraten. Anhand dieses Gitternetzes und des Koordinatensystems erstellen sie einen Plan für ihr Labyrinth.

Nach der Planungsphase basteln die SchülerInnen ihre Labyrinthe. Ist dieser Schritt abgeschlossen, kommt die größte Herausforderung: das Programmieren.

Dazu öffnen sie die Internetseite www.ozoblockly.com und klicken dort auf "Get Started". Das Programm öffnet sich selbstständig im Browser. Es ist möglich, Ozoblockly auf dem Smartphone, Tablet, Laptop oder Desktop zu nutzen.

Für diese Unterrichtseinheit empfiehlt es sich, Ozoblockly mindestens auf den Schwierigkeitsgrad Advanced *(Mode 4)* einzustellen. Hier finden sich unter dem Tab "*Movement*" alle Blöcke, die benötigt werden, um durch das Labyrinth zu navigieren.

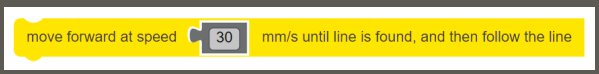


Vertiefung:

- Versuche noch andere Befehle in den Code zu integrieren.
- Diskutiere im Plenum über die Schwierigkeiten und die verschiedenen Lösungsansätze.

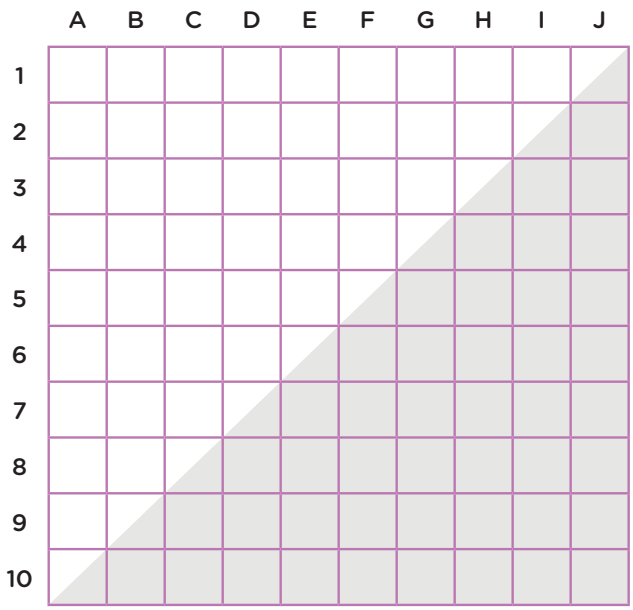
Tipps und Tricks:

Tipp: An schwierigen Stellen *(zum Beispiel Klopapierrolle als Tunnel)* hilft oft ein schwarzer Strich als Orientierungslinie, damit der Roboter nicht hängen bleibt.



Die Dimensionen des Gitternetzes können auch kleiner gewählt werden.

Material	Ort (z.B. A1-D1)



Slotcar/

Material

- Ozobot
- Blatt: SlotCarChallenge
- Tablet oder Computer

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Bezug zum Lebensalltag

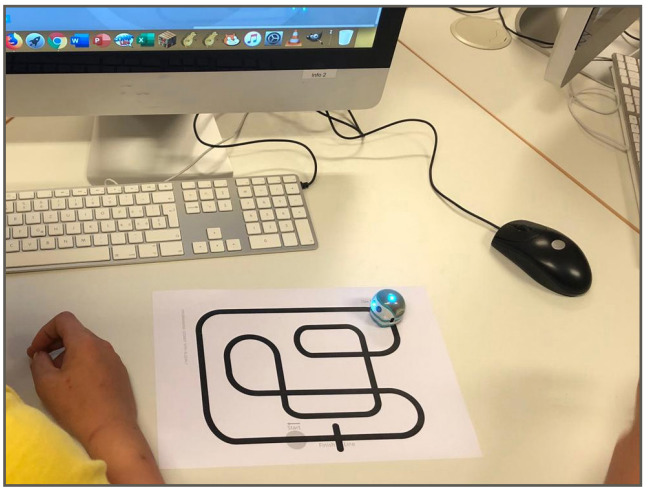
Funktionen werden beim Programmieren oft gebraucht. Sie erlauben, wiederkehrende Programmierblöcke abzuspeichern und diese aufzurufen und durchzuführen.

Ablauf

Für die folgenden Übungen wird in der Ozoblockly-Programmsoftware die Stufe Intermediate *(Mode 3) (oder höher)* eingestellt. Als Training programmieren die SchülerInnen den Ozobot so, dass er den Parcours einmal abfährt. Der Ozobot muss an jeder Kreuzung geradeaus fahren. Da es sich um ein Rennen handelt, soll der Ozobot mit seiner größten Geschwindigkeit fahren.

Beim eigentlichen Rennen fährt der Ozobot drei Runden. Die SchülerInnen passen das vorherige Programm an. Dafür gibt es mehrere Möglichkeiten. Die Lehrperson weist die SchülerInnen darauf hin, dass es sinnvoll ist, das Programm so kurz wie möglich zu halten. Dazu eignen sich unter anderem die *Wiederholen*-Schleifen.

Zum Schluss soll der Roboter die drei Runden abfahren und bei jeder Runde die Farbe wechseln. Die Lehrperson kann bei den beiden letzten Beispielen den SchülerInnen die Funktionsblöcke in Ozoblockly zeigen und erklären.



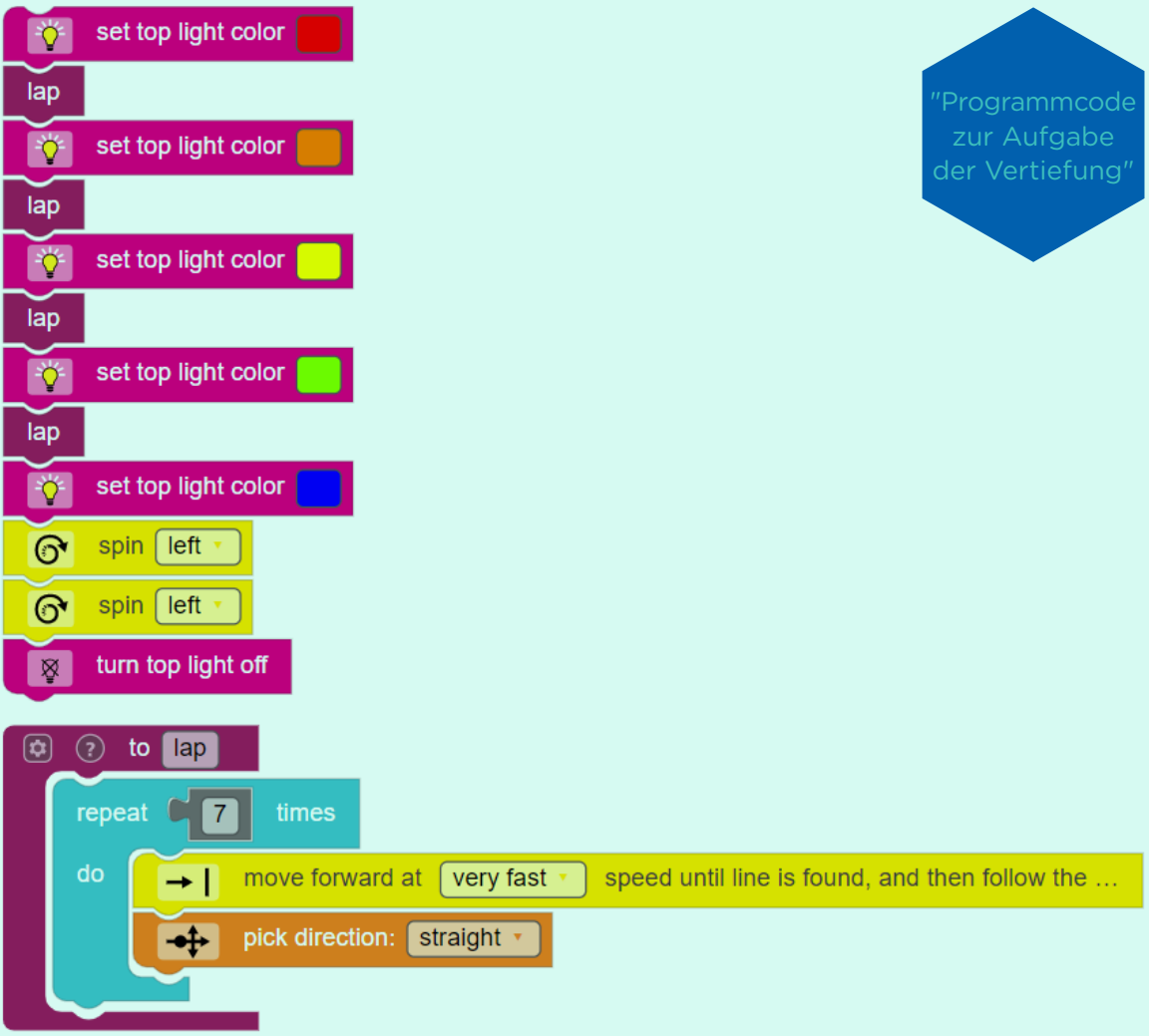
Vertiefung:

- Nach der fünften Runde soll der Ozobot eine ganze Drehung ausführen und sich danach ausschalten.
- Die SchülerInnen stellen sich gegenseitig weitere Aufgaben und kontrollieren gemeinsam die Programmierung mit einem Testlauf.

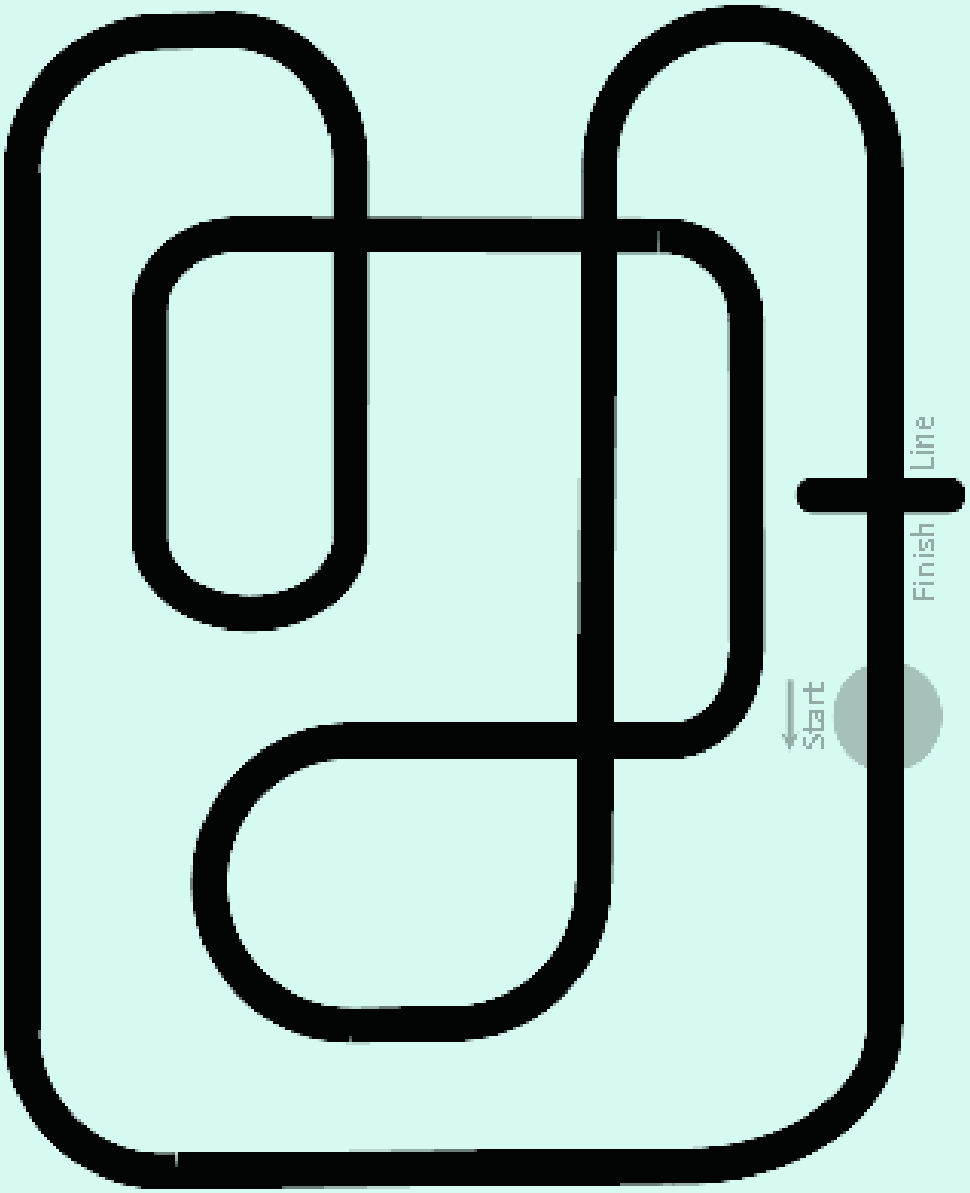
Tipps und Tricks:

Die Funktionsblöcke können zuerst *unplugged* eingeführt werden. Die SchülerInnen spielen Roboter. Einige Befehle werden als Funktionsblöcke abgespeichert und können wiederholt aufgerufen werden.

Beispiel: Funktion *Treppenstufe*: gehe 1 Schritt vorwärts – drehe dich um 90° nach rechts – gehe 1 Schritt vorwärts – drehe dich um 90° nach links. Ruft man die Funktion *Treppenstufe* mehrfach auf, bewegt sich der Roboter treppenförmig weiter.



"Programmcode zur Aufgabe der Vertiefung"



* *"Quelle: https://ozoblockly.com/resource/challenges/Slot-Car-Track.pdf"*

Vielecke/

Material

- Ozobot
- Tablet oder Computer
- Internetseite https://games.ozoblockly.com/

Bezug zum Lehrplan

Mathématiques : Espace et formes : Analyser et représenter des figures géométriques dans le plan et l'espace

Bezug zum Lebensalltag

Die SchülerInnen befassen sich mit den Eigenschaften der Vielecke: Anzahl der Seiten, Parallelen, Innenwinkel, Außenwinkel, Winkelsummen...

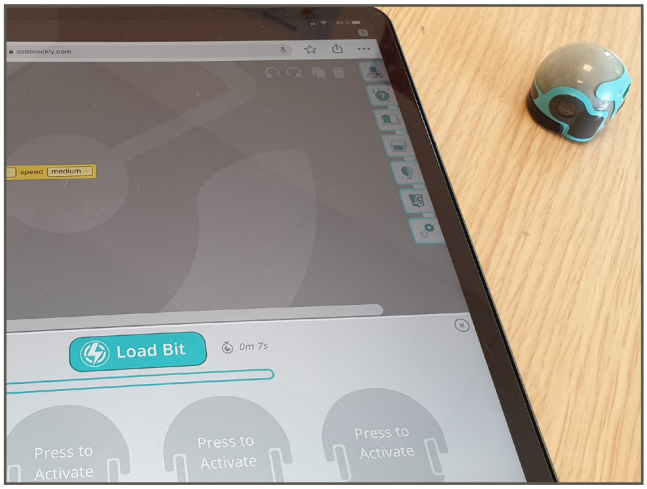
Ablauf

Die SchülerInnen erhalten pro Gruppe einen Ozobot und ein Tablet.

Die erste Aufgabe besteht darin, den Ozobot mit der Webapp Ozoblockly so zu programmieren, dass der Roboter ein Quadrat abfährt.

Nachdem die SchülerInnen diese Aufgabe erfolgreich gelöst haben, widmen sie sich dem Programmieren von anderen geometrischen Formen. Je nach gewünschtem Schwierigkeitsgrad laufen der Roboter ein Dreieck oder ein Fünfeck ab. Beim Dreieck und Fünfeck müssen die SchülerInnen sich mit den Winkelsummen *(Innen- oder Außenwinkel)* dieser geometrischen Figuren befassen.

Es gibt mehrere zielführende Lösungswege. Bei einem falschen Lösungsweg fährt der Roboter nicht die gewünschte Form ab, und die SchülerInnen erkennen sofort, dass die Programmierschritte angepasst werden müssen.



Vertiefung:

Die SchülerInnen lassen den Ozobot einen Kreis mit einer vorgegebenen Radius- oder Diametergröße fahren.

Tipps und Tricks:

Webseite zum Programmieren des Ozobots: https://ozoblockly.com/

Stadtplan/

Material

- Ozobot

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Langues : Compréhension écrite : Repérer les informations d'un texte et les exploiter

Bezug zum Lebensalltag

Bei dieser Unterrichtseinheit wird neben dem Textverständnis auch das algorithmische Denken gefördert. Die SchülerInnen berücksichtigen die Reihenfolge der besichtigten Orte genau und programmieren den Ozobot entsprechend dieser Handlungsabfolge.

Ablauf

Die SchülerInnen erhalten eine Geschichte und einen Stadtplan. Sie lesen die Geschichte durch, arbeiten die verschiedenen Etappen in entsprechender Reihenfolge aus. Anschließend programmieren sie den Ozobot über ozoblockly.com so, dass er ohne Farbcodes die Geschichte chronologisch korrekt auf dem Stadtplan "abfährt".

Beispiel einer Geschichte:

„Tom, denke bitte daran, das Paket an der Packup-Station abzuholen", ruft sein Vater, bevor er zur Arbeit fährt. „Geht in Ordnung!", antwortet Tom.

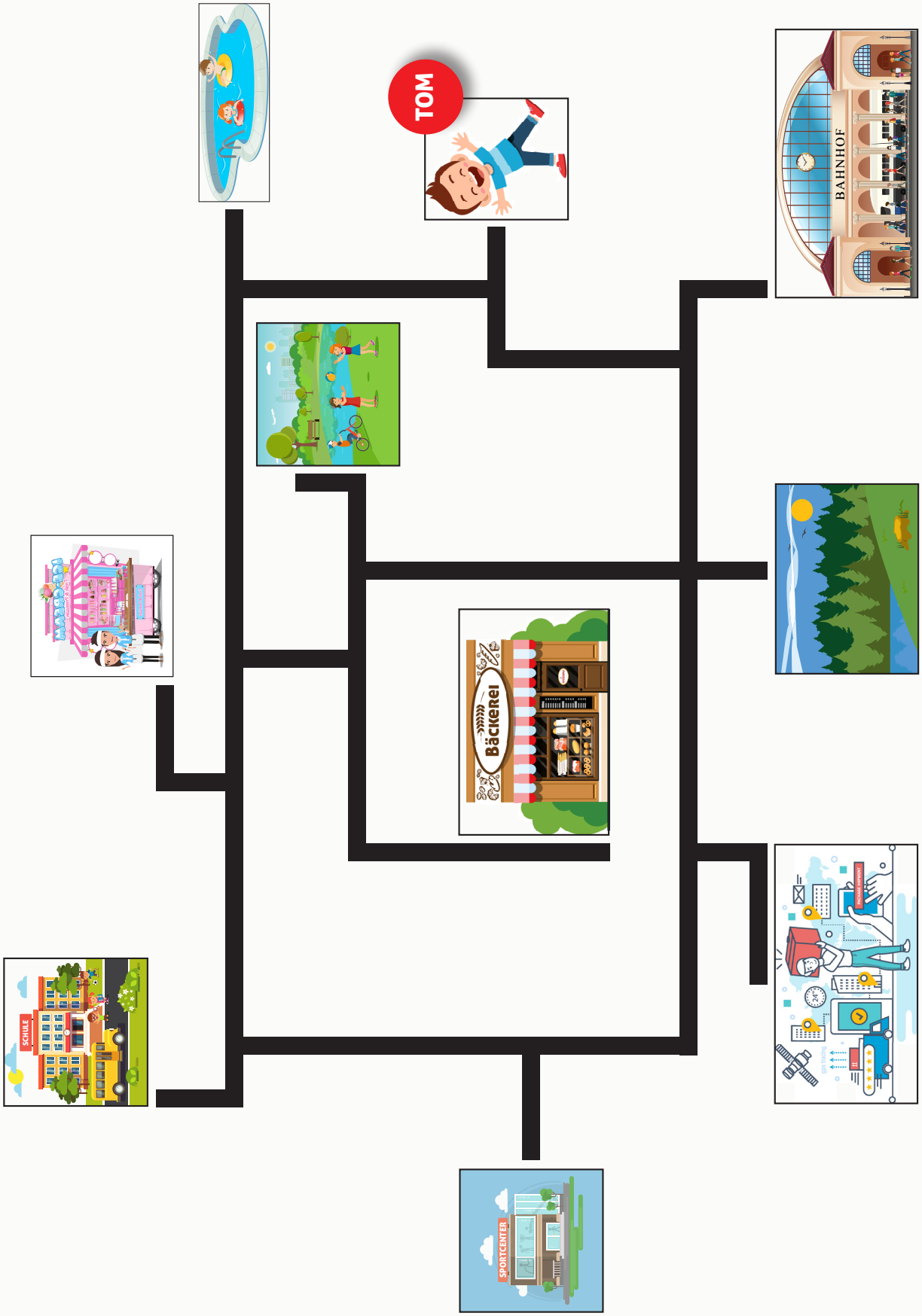
Er schwingt sich auf sein Fahrrad. „Das Paket hole ich erst nach dem Schwimmen ab", denkt er unterwegs und radelt zuerst zum Schwimmbad.

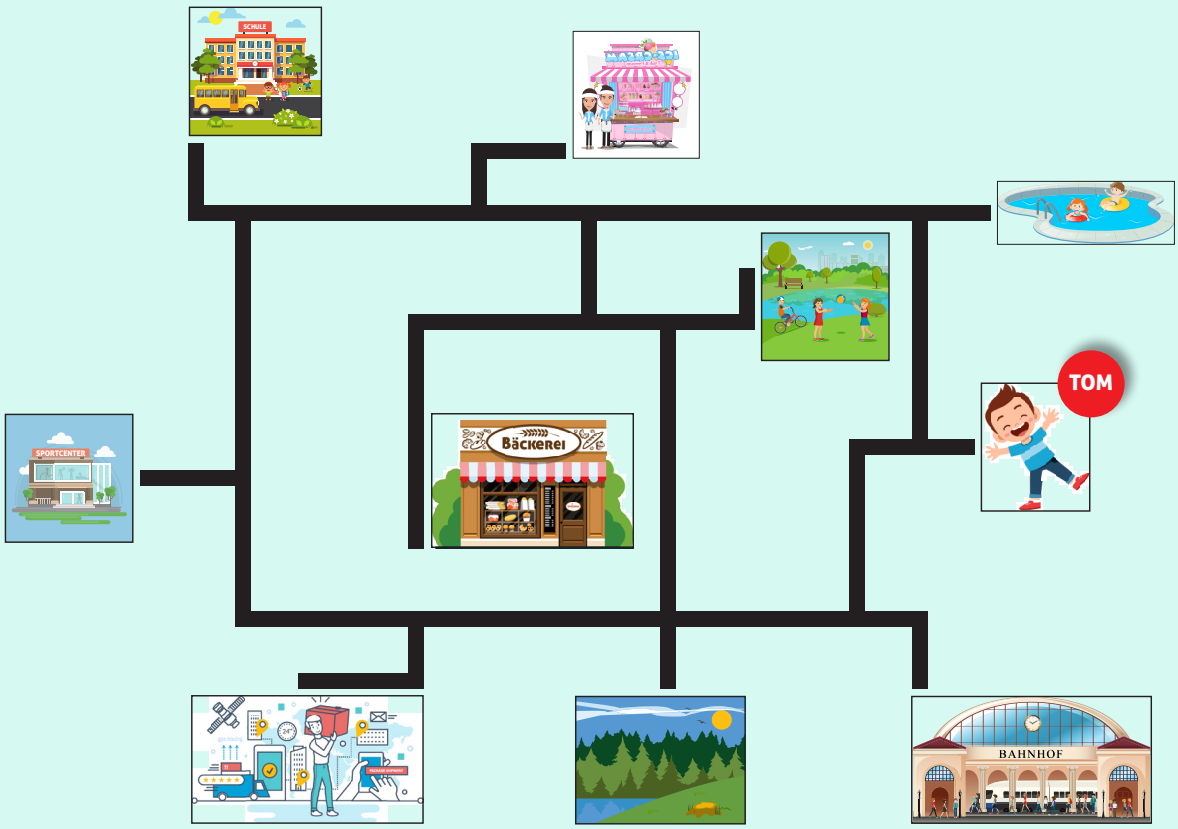
Nach dem Schwimmen ist Tom hungrig. Er fährt deswegen zuerst zum Bäcker und kauft sich eine Butterbrezel.

„Soll ich nun durch den Park fahren oder am Sportzentrum vorbei?", fragt sich Tom, als er die Bäckerei verlässt.

Da er noch seine Freundin Elisa treffen möchte, die zurzeit im Sportzentrum trainiert, entscheidet er sich für diesen Weg.

Gemeinsam fahren Elisa und Tom dann zur Pack-up-Station. Hier verabschiedet sich Elisa, und Tom fährt auf dem kürzesten Weg nach Hause.


TOM
BAHNHOF
Bäckerei
SCHULE
SPORTCENTER



Vertiefung:

Diese Aufgabe lässt sich leicht differenzieren, indem die Geschichte schwieriger oder einfacher gestaltet wird.

Tipps und Tricks:

Die SchülerInnen entwerfen selbst einen Stadtplan, und ihre KameradInnen schreiben Geschichten dazu.

Vergrößern Sie die Aufgabenkarten auf das A3-Format, und zeichnen Sie gegebenenfalls die Linien mit dem Ozobot-Stift nach.

Scratch

Koordinaten-system

Material

- Tablet oder Laptop
- Scratch

Bezug zum Lehrplan

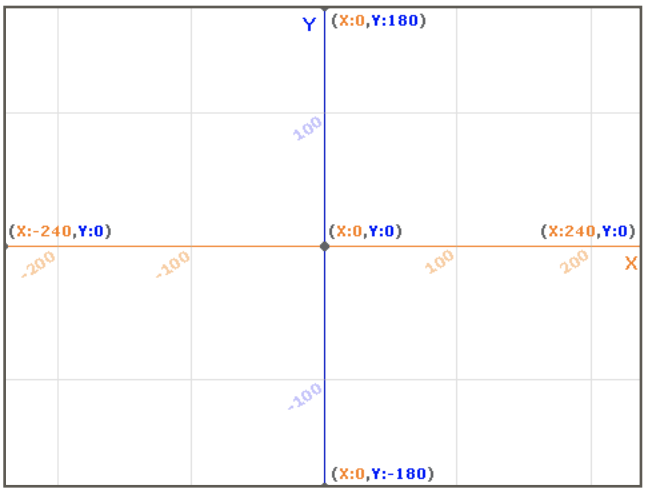
Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Bezug zum Lebensalltag

Zum genauen Platzieren von Objekten arbeiten Computer oft mit einem Koordinatensystem. Jeder Punkt auf dem Bildschirm entspricht einer Koordinate und kann so einzeln angesprochen werden.

Ablauf

Scratch positioniert die Elemente auf der Bühne nach dem folgenden Koordinatensystem:



Der Punkt (0,0) liegt in der Mitte der Bühne. Die X-Koordinaten reichen von -240 bis 240, die Y-Koordinaten von -180 bis 180.

Die SchülerInnen positionieren die Katze an verschiedenen Stellen auf der Bühne. Unterhalb der Bühne *(siehe gelb markierte Zahlen auf der Abbildung)* sehen sie, an welcher Position die Figur steht.

Hier befindet sich die Katze an der Position X:121 und Y:55

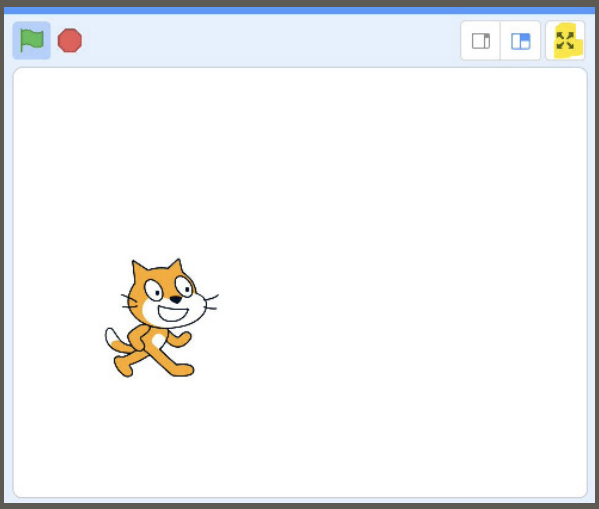
Die SchülerInnen arbeiten zu zweit. Eine/r gibt dem/der PartnerIn eine Position an, der/die andere bewegt die Katze dorthin.

Zur Abwechslung wählen die SchülerInnen rechts unten ein Bühnenbild aus. Nun beschreiben die SchülerInnen, welche Elemente des Bühnenbildes sich an welcher Position befinden. Zur Kontrolle bewegen sie die Katze an die entsprechende Position. Anstelle der Katze können auch andere Figuren ausgewählt werden. Die Größe der Figur ist ebenfalls veränderbar.

Vertiefung:

Die SchülerInnen schreiben folgendes Skript.

Sie vergrößern das Fenster und klicken auf den grünen Pfeil.



Die Katze bewegt sich auf dem Bildschirm. Die SchülerInnen versuchen festzustellen, ob die Koordinaten X und Y der Position der Katze positiv oder negativ sind.

Tipps und Tricks:

Im Skript kann die Wartezeit um 10 Sekunden verändert werden. Ist die Wartezeit kürzer, so müssen die SchülerInnen schneller reagieren, um die Position zu bestimmen. Auf diese Weise lässt sich diese Aufgabe differenziert einsetzen.

Maussteuerung und Katzenbilder/

Material

- Tablet oder Laptop
- Scratch

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Bezug zum Lebensalltag

Es gibt Teile einer Programmierung, die oft wiederholt werden. Das kann eine Hintergrundmusik sein oder eine Animation. Für diesen Fall sind in Programmiersprachen Wiederholungen *(Loops)* vorgesehen. Loops können fortlaufend wiederholt werden, so lange, bis das Programm zu Ende ist.

Ablauf

1. *Ein neues Projekt öffnen und mit dem Namen "Maussteuerung" benennen.*
2. *Im Programmierbereich auf die Figuren klicken. Die SchülerInnen wählen sich eine Figur aus.*
3. *Die SchülerInnen nehmen aus der Blockpalette "Ereignisse" den Steuerungsbefehl "Wenn grüne Fahne angeklickt" heraus. Anschließend ziehen die SchülerInnen den Block "Gehe zu Mauszeiger" in den Skriptbereich.*
4. *Sie nehmen aus der Blockpalette den Steuerungsbefehl "Wiederhole fortlaufend". Diese Schleife umschließt den Block "Gehe zu Mauszeiger" (siehe Abbildung rechts).*
5. *In Scratch können auf vielfältige Weisen Kunstwerke angefertigt werden. Die SchülerInnen ziehen in der Blockpalette "Aussehen" den Block "Ändere Effekt Farbe um (25)" in die Schleife. Zusätzlich ergänzen sie die Schleife mit dem Block „Erzeuge Klon von mir selbst". Dieser Block stempelt überall, wo die Katze hinkommt, ein Bild der Figur auf die Bühne.*

Vertiefung:

- Die SchülerInnen verändern auch die Größe der Figur.
- Die SchülerInnen wählen unterschiedliche Zahlen und Einstellungen aus. Die Farb-Effekte müssen nicht bei 25 eingestellt werden. Je kleiner die Zahl, desto langsamer verändert die Katze ihre Farbe, ähnlich wie bei einem Regenbogen.
- Neben den Farbänderungen gibt es noch andere Effekte, die in den Blockbereichen getestet werden können.
- Die SchülerInnen steuern die Effekte besser und genauer, wenn eine beliebige Taste als Auslöser des Skripts verwendet wird.

Tipps und Tricks:

Beim Klicken auf die blaue Schaltfläche links über der Bühne werden die Skripte ausgeblendet, und es ist nur noch das ablaufende Programm zu sehen. Zum Verkleinern der Bühne wird dieselbe Schaltfläche nochmals gedrückt.

Nachrichten versenden

Material

- Tablet oder Computer
- Scratch

Bezug zum Lehrplan

Langues : Production écrite : Rédiger différents types de textes

Bezug zum Lebensalltag

Computerprogramme laufen nicht immer linear ab, das heißt ohne Unterbrechung von Anfang bis Ende. Sie müssen auch auf Eingaben oder andere Ereignisse reagieren können: zum Beispiel auf Werte von Sensoren, die übermittelt werden.

In dieser Unterrichtseinheit lernen die SchülerInnen, wie sie verschiedene Programmteile erst dann starten, wenn ein Ereignis *(in diesem Fall der Empfang einer Nachricht)* eintritt.

Ablauf

1. *Ein neues Projekt öffnen und mit dem Namen "Hallo" benennen.*
2. *Im Programmierbereich auf die Figuren klicken. Die SchülerInnen wählen sich eine Figur aus.*
3. *Die SchülerInnen nehmen aus der Blockpalette "Ereignisse" folgenden Steuerungsbefehl heraus: "Wenn grüne Fahne angeklickt". Anschließend bewegen sie den Block "Sage Hallo für 2 Sekunden" in den Skriptbereich. Sie füllen das Feld mit einer eigenen Nachricht aus.*

Vertiefung:

Die SchülerInnen können mehrere Figuren miteinander sprechen lassen *(Beispiel: ein Dialog oder ein Katz-und-Maus-Spiel)*.

Dazu müssen sie die Abfolge des Gespräches programmieren. Nach dem Block "Wenn ich... empfange" können die SchülerInnen unterschiedliche Abfolgen selbst programmieren *(die andere Figur könnte zum Beispiel weitergehen)*.

Tipps und Tricks:

Texte aus Schulbüchern können auf diese Art und Weise von den SchülerInnen in Dialoge umgewandelt werden.

Labyrinth/

Material

- Tablet oder Laptop
- Scratch

Bezug zum Lehrplan

Mathématiques : Espace et formes : S'orienter dans le plan et l'espace

Bezug zum Lebensalltag

Für Computer ist es wichtig, ganz genaue Befehle zu erhalten, denn nur diese führt er aus.

In dieser Unterrichtseinheit lernen die SchülerInnen die Wichtigkeit einer präzisen Beschreibung. Sie lernen, sich innerhalb eines Koordinatensystems zu bewegen.

Ablauf

Auffrischung X: links/rechts Y: hoch/runter

X: rechts benutzen wir eine positive Zahl

X: links benutzen wir eine negative Zahl

Y: hoch benutzen wir eine positive Zahl

Y: runter benutzen wir eine negative Zahl

Nachdem die SchülerInnen gelernt haben, sich mithilfe der Koordinaten weiter zu bewegen, nutzen sie dieses Wissen, um Scratch oder eine andere Figur durch ein Labyrinth zu schicken.

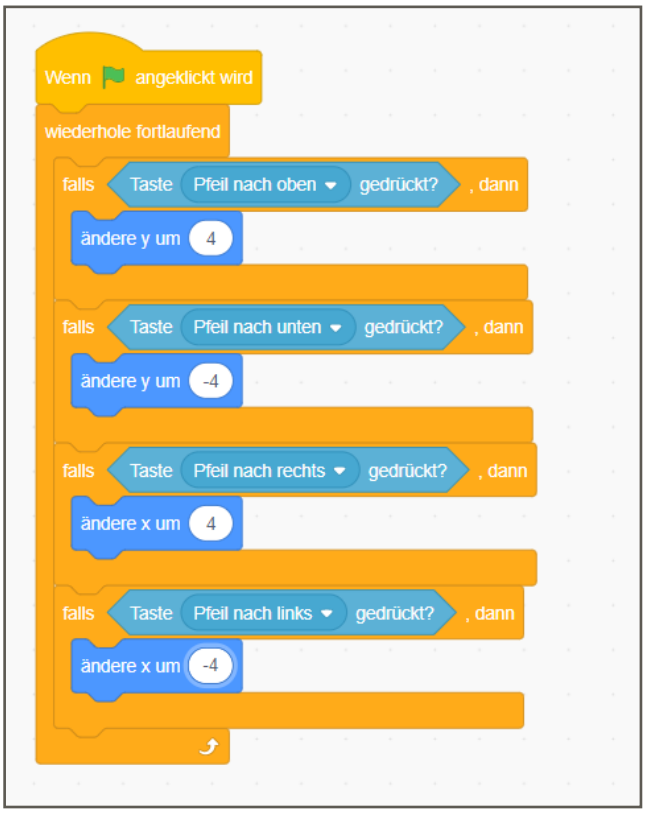
Rufen Sie die Internetseite https://scratch.mit.edu auf, und klicken Sie auf "Beginne mit dem Erstellen".

1. Bewegungen der Figur

Wenn das Programm startet, prüft es fortlaufend, ob die "Pfeiltaste nach oben" gedrückt wurde. Wird die Taste nicht gedrückt, passiert nichts. Wird die Taste gedrückt, so ändert sich die Y-Position der Figur um 4. Da sie sich um eine positive Zahl verändert, bewegt sie sich nach oben.

Da die Strecke des Labyrinthes nicht gerade ist, brauchen die SchülerInnen ebenfalls die "Linke-", die "Rechte-" und auch die "Nach unten-Pfeiltaste". Die SchülerInnen sollen an die positiven und negativen Zahlen denken und daran, dass die X-Achse den horizontalen Bewegungen entspricht.

Der Code sieht jetzt folgendermaßen aus:



Die SchülerInnen testen ihren Code. Macht die Figur das, was sie sollte? Wenn nicht: Was macht sie? Die SchülerInnen finden eine Lösung, indem sie die einzelnen Blöcke genau untersuchen.

Die SchülerInnen speichern ihr Projekt ab.

2. Labyrinth

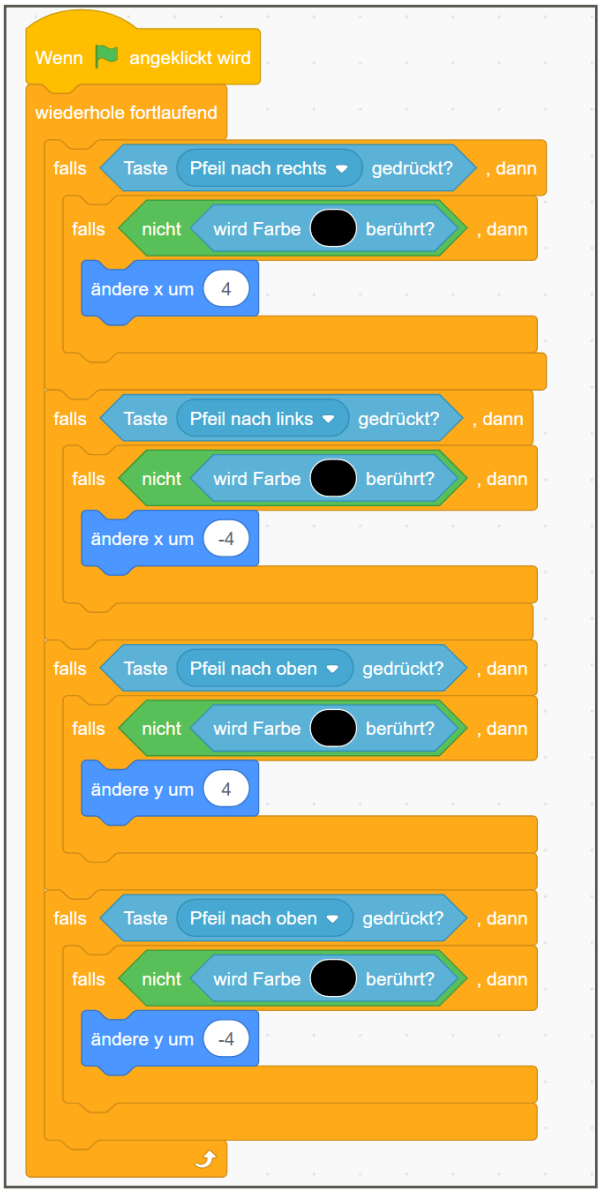
Die SchülerInnen erstellen ein Labyrinth. Dazu benötigen sie ein Blatt Papier und einen Stift. Die SchülerInnen denken sich in ihrer Gruppe ein Labyrinth aus und malen es auf. Erst danach wird das Labyrinth in Scratch nachgezeichnet.

Hierzu wird ein neues Bühnenbild erstellt. Dieses Bild wird *"Labyrinth"* genannt und stellt den Hintergrund dar.

Wenn sich die Figur durch das Labyrinth bewegt, werden die SchülerInnen feststellen, dass sie durch Mauern gehen kann. Das muss natürlich verhindert werden.

Deshalb müssen die SchülerInnen in dem Programm vorhersehen, dass etwas anderes passiert, wenn die Figur die Wand *(also die Farbe der Mauern des Labyrinthes)* berührt. Hierzu benötigen sie den "Wird... berührt?"-Block.

Nur wenn die Figur die Mauern des Labyrinthes *(also die Farbe der Mauern - in diesem Beispiel die Farbe Schwarz)* NICHT berührt, soll sie sich in die entsprechende Richtung bewegen.

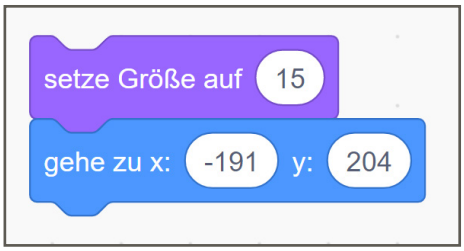


Fast geschafft! Nur ist die Figur noch zu groß, und sie weiß nicht, wo sie anfangen soll und wo das Ende des Labyrinthes ist.

3. Position und Größe der Figur festlegen

Zuerst legen die SchülerInnen die Startposition und die Größe fest. Diese Zahlen innerhalb beider Blöcke sind abhängig von jedem Labyrinth.

Folgende Blöcke können zum Beispiel benutzt werden:



Jetzt fehlt nur noch das Ende. Das Programm muss wissen, wann das Ziel erreicht ist. Hierzu eignet sich ein Objekt am Ende des Labyrinths. Also erschaffen die SchülerInnen eine neue Figur oder wählen eine bereits vorgefertigte aus der Bibliothek aus.

In diesem Beispiel wurde eine Maus ausgewählt. Genau wie bei der Katze legen die SchülerInnen zuerst die Größe und Position der Maus fest. Sie wählen die Maus aus und geben folgenden Code für die Maus ein:

Tipp: Die SchülerInnen müssen darauf achten, dass die Figur der Maus ausgewählt ist. Jetzt legen sie noch fest, was passiert, wenn die Figur *(die Katze)* die Maus fängt. Hierzu soll das Programm wieder fortlaufend prüfen, ob die Maus berührt wird.

Achtung: Die SchülerInnen müssen wieder die Katze auswählen!

Genau wie bei dem Labyrinth: "Falls Maus berührt? dann..."

Vertiefung:

Weiterführend können noch weitere Schwierigkeitsgrade hinzugefügt werden.

Tipps und Tricks:

Bei dieser Applikation handelt es sich bereits um eine komplexere Aufgabe. Wichtig ist, die Aufgabe zuerst in Einzelschritte zu zerlegen und nach und nach die einzelnen Komponenten zu programmieren.

Es gibt noch andere Möglichkeiten, den Algorithmus des Programmes aufzuschreiben. So kann zum Beispiel die Bewegung der Katze nach rechts auch folgendermaßen programmiert werden:

Kniwwelino

Kniwwelino
Kniwwelino hinzufügen
Quellcode
wiederholend
A
B
Kniwwelino

Stoppuhr/

Material

- Kniwwelino
- USB Kabel *(+ Powerbanks)*
- Tablet oder Computer
- Externe LED-Lampen

Bezug zum Lehrplan

Mathématiques : Résolution de problèmes d'arithmétique : Interpréter et évaluer les informations et les résultats

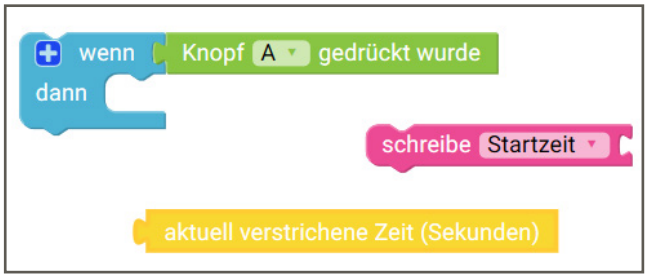
Bezug zum Lebensalltag

In dieser Unterrichtseinheit bauen die SchülerInnen ihre eigene Stoppuhr, die in der Klasse eingesetzt werden kann. Mit dem Knopf A wird die Stoppuhr gestartet und mit dem Knopf B gestoppt. Die Matrix zeigt die verstrichene Zeit an.

Ablauf

Die SchülerInnen verbinden ihr Kniwwelino mit dem WLAN und öffnen im Browser die Seite: http://code.kniwwelino.lu.

Wenn Knopf A gedrückt wird, soll die verstrichene Zeit in einer Variablen *(hier Startzeit)* gespeichert werden. Dazu werden folgende Blöcke benötigt:



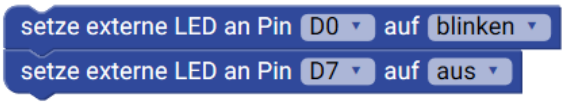
Wenn Knopf B gedrückt wird, soll die verstrichene Zeit in einer Variablen *(hier Endzeit)* gespeichert werden. Die Differenz von Endzeit und Startzeit wird errechnet. Dieser Wert wird auf der Matrix ausgegeben. Dazu werden folgende Blöcke benötigt:

Wenn Knopf A gedrückt wird, soll die Matrix wieder gelöscht werden. Hierzu braucht es folgenden Block:

Vertiefung:

An diese Stoppuhr können nach und nach weitere Funktionen, wie zum Beispiel Lichtsignale, mit angebunden werden: Beim Starten der Stoppuhr soll eine grüne Lampe blinken. Wenn die Uhr stoppt, leuchtet eine rote LED-Leuchte.
→

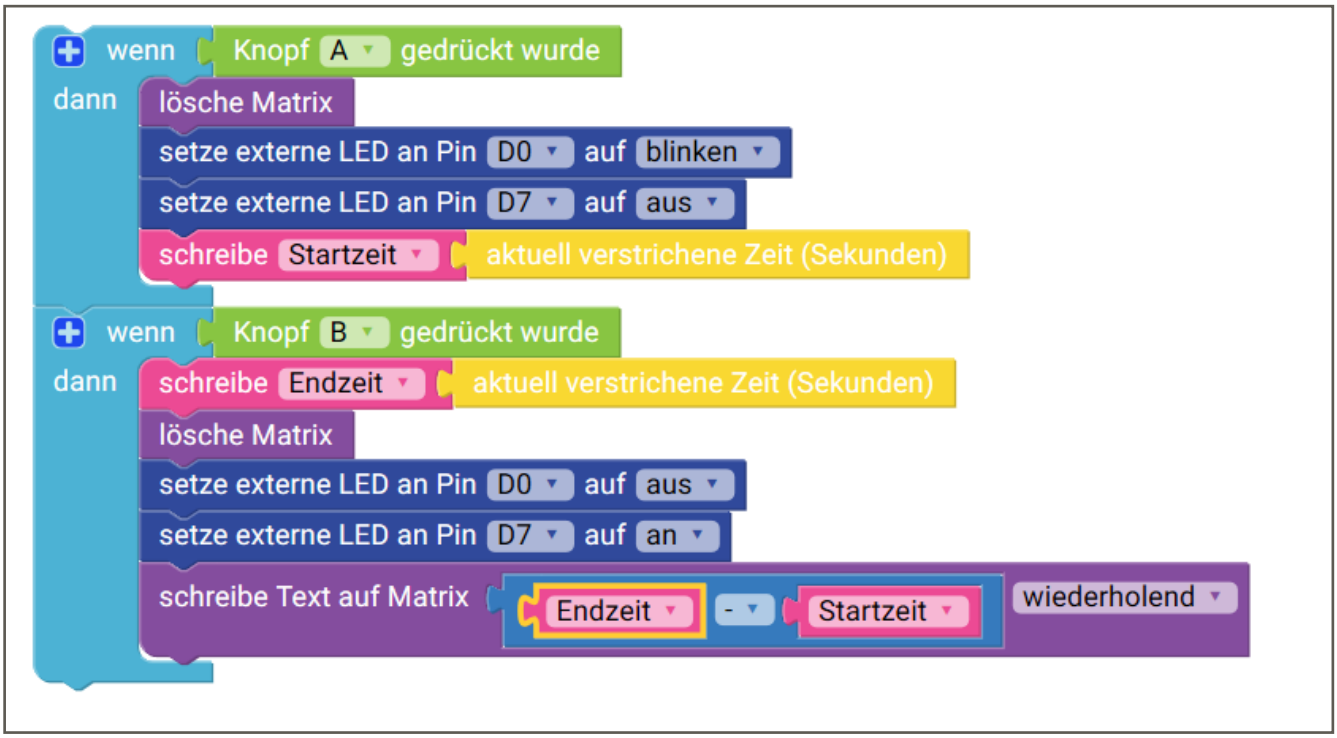
Benutze hierzu folgende Blöcke und setze sie an die richtigen Stellen. Denke daran, dass es nicht reicht, der Lampe den Befehl zu geben sich einzuschalten. Wenn du willst, dass sie sich bei einem bestimmten Knopfdruck ausschaltet, musst du dies in deinem Code vorsehen.



Tipps und Tricks:

Es ist empfehlenswert, immer nur eine neue Funktion hinzuzufügen und umgehend den neuen Code zu testen. So fällt die Fehlersuche leichter.

Mögliche Lösung:



Happy Birthday/

Material

- Kniwwelino
- 2 Krokodilklemmen
- Tablet oder Computer

Bezug zum Lehrplan

L'expression créative, l'éveil à l'esthétique et à la culture dans le domaine de la musique : Percevoir la musique

Bezug zum Lebensalltag

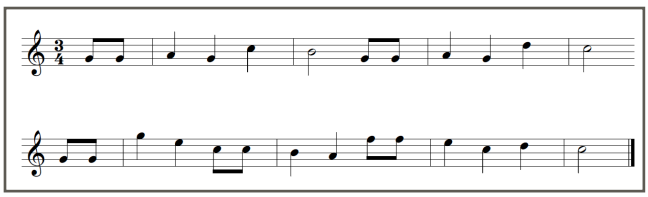
In den 1990er Jahren wurden Buzzer *(Summer)* in Computer und Spielkonsolen eingebaut, um Töne von sich geben zu können. Die wichtigsten Elemente eines Piezo-Buzzers sind das Metallgehäuse und die kleine Keramikscheibe, die in Schwingung versetzt wird.

In dieser Unterrichtseinheit wird anhand des Buzzers ein Lied abgespielt.

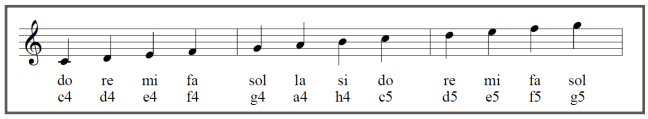
Dabei werden die Noten kodiert. Computer arbeiten oft mit Codes. Die Noten sind in diesem Fall auch Codes.

Ablauf

Das Geburtstagslied soll für Kniwwelino kodiert werden. Hier ist die Melodie:



Die Kodierung entspricht den englischsprachigen Notennamen. Jede Note entspricht einem Buchstaben mit einer Ziffer. Dabei zählt nur der Notenkopf, der dicke schwarze Punkt.



Die SchülerInnen erhalten die Noten des Liedes und schreiben die Notencodierungen unter die Noten.

Alle schwarzen Noten mit einem einfachen Strich *(nach oben oder unten)* haben die Länge 1/4, die Noten, die jeweils mit einem Balken verbunden sind, haben die Länge 1/8.

Die weißen Noten haben die Länge 1/2.

Nun ergänzen die SchülerInnen die Notencodierung des Liedes mit den entsprechenden Werten.

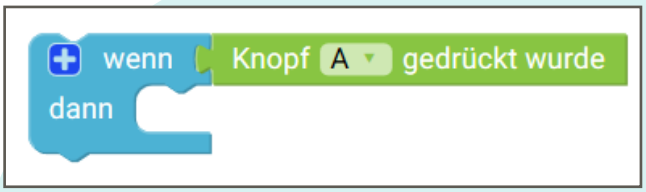
Sie nehmen dann den Kniwwelino, zwei Krokodilklemmen und den Piezo-Buzzer. Diesen schließen sie an das Kniwwelino-Board. Anschließend rufen sie die Internetseite https://code.kniwwelino.lu auf und stellen das Programm auf "Erweitert“ ein.

Durch den Buzzer kann der Kniwwelino Lieder abspielen. Dazu werden verschiedene Noten in einer bestimmten Länge abgespielt.

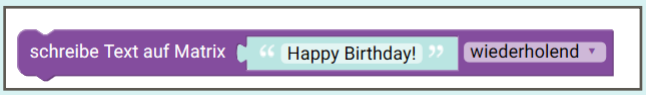
Folgende Blöcke werden gebraucht:

Das Lied wird nur gespielt, wenn Knopf A gedrückt wird. Auf der Matrix soll *„Happy Birthday“* erscheinen.

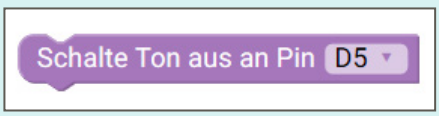
Dazu werden folgende Blöcke gebraucht:



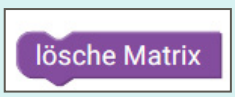
Wenn Knopf A gedrückt wird dann...



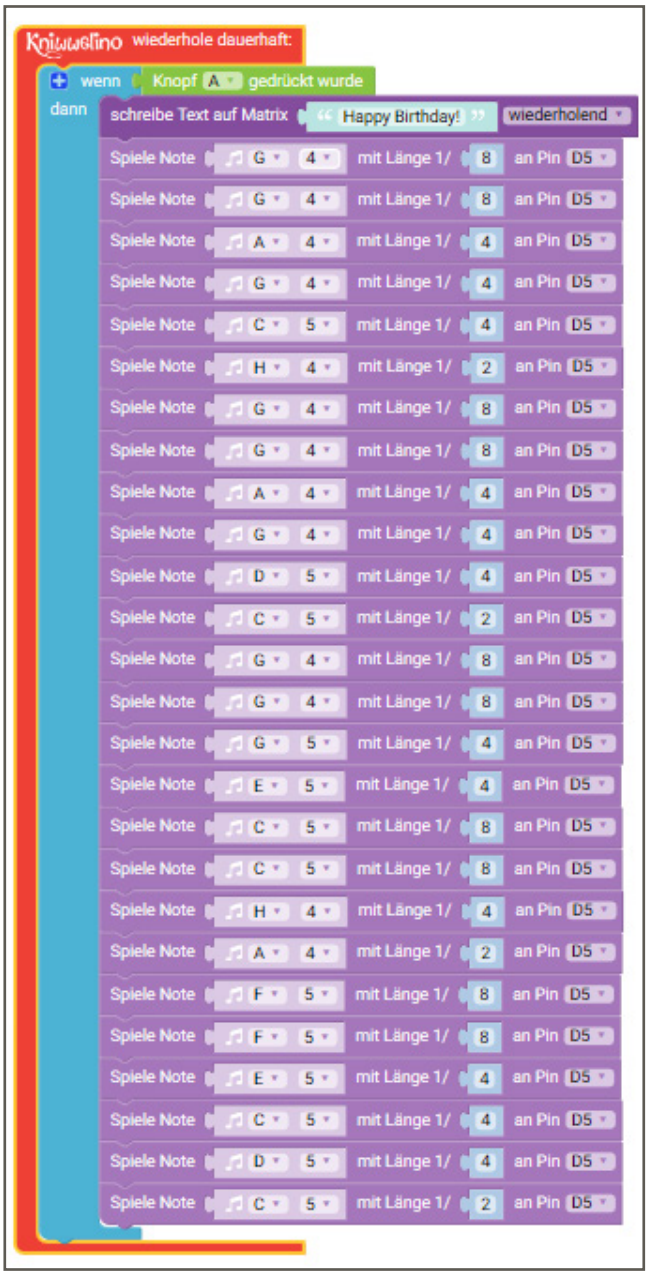
→ Schreibe Happy Birthday auf die Matrix



→ Ton ausschalten



→ Text löschen



Vertiefung:

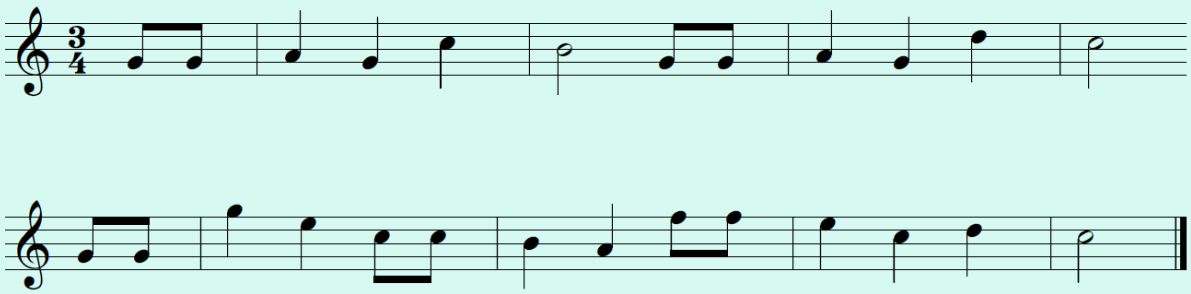
Die SchülerInnen suchen sich andere Lieder zum Abspielen aus. Im Internet finden sich viele Piezo-Buzzer-Arduino-Tutorials mit fertigen Liedern.

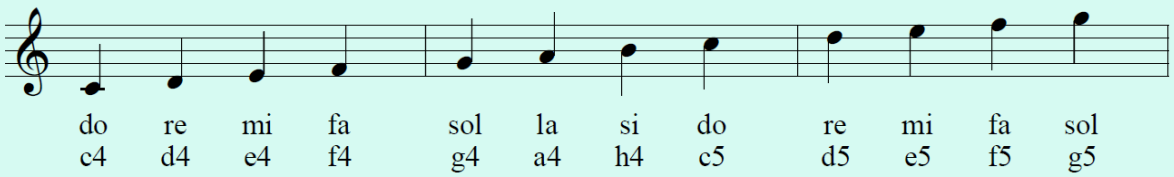
Tipps und Tricks:

Überprüfen Sie, ob der richtige Anschluss ausgewählt wurde. Der Piezo-Buzzer wird in diesem Beispiel an Pin D5 angeschlossen.

Nicht vergessen, am Ende der Melodie die Blöcke "Ton aus“ und "Lösche Matrix“ zu benutzen.

Da die Notenlängen den Brüchen 1/2, 1/4, 1/8 entsprechen, eignen sie sich auch, Additionen mit Brüchen zu üben. Beispiel: Wie viele Achtelnoten passen in eine halbe Note?




do re mi fa sol la si do re mi fa sol
c4 d4 e4 f4 g4 a4 h4 c5 d5 e5 f5 g5

Kniwwelino Matrixanzeige

Material

- Kniwwelino *(empfohlen: jeweils 1 Kniwwelino für 2 Lernende)*
- Tablet oder Computer
- Steckdosenleiste, Ladekabel oder Powerbank

Bezug zum Lehrplan

Langues : Compréhension de l'écrit : Analyser, comparer et évaluer des textes

Bezug zum Lebensalltag

Bei einer Fahrt auf der Autobahn fallen sicherlich die elektronischen Hinweistafeln auf, die vor Gefahren warnen sollen. "Unfall in 4 km“ oder "Achtung! Glatteis!“ leuchtet es in riesigen Buchstaben auf den Anzeigetafeln.

Wie funktionieren diese Tafeln? Wie werden die Buchstaben dargestellt und wie werden die Hinweise geändert? Was ist eine Matrixanzeige?

Auf einige dieser Fragen kann diese Unterrichtseinheit Antworten geben, denn die SchülerInnen lernen, wie eine Matrixanzeige funktioniert und wie diese verändert werden kann.

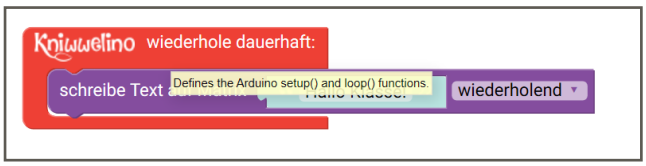
Ablauf

Die ersten Schritte mit der Klasse können sich auf den spielerischen, explorativen Umgang mit visuellem Code auf https://code.kniwwelino.lu beschränken.

Nachdem die Kniwwelino-Platine mit dem WLAN verbunden und in der Programmierumgebung zu der Liste der Kniwwelinos hinzugefügt ist, verändern die SchülerInnen die vorgegebenen Blöcke auf Kniwwelino Blockly beliebig und lassen diese dann ausführen.

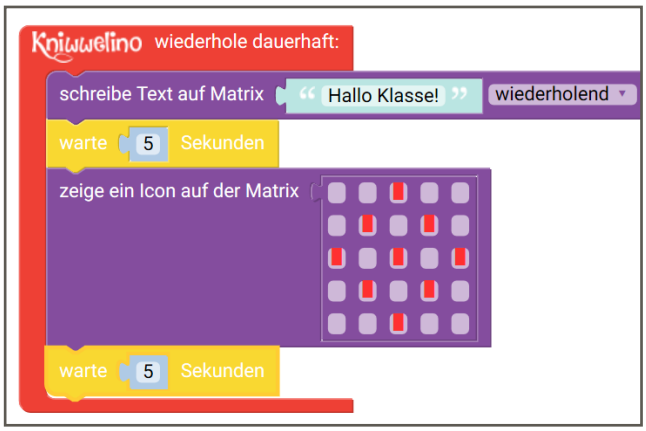
Beispiele

- Eigenen Namen eingeben.

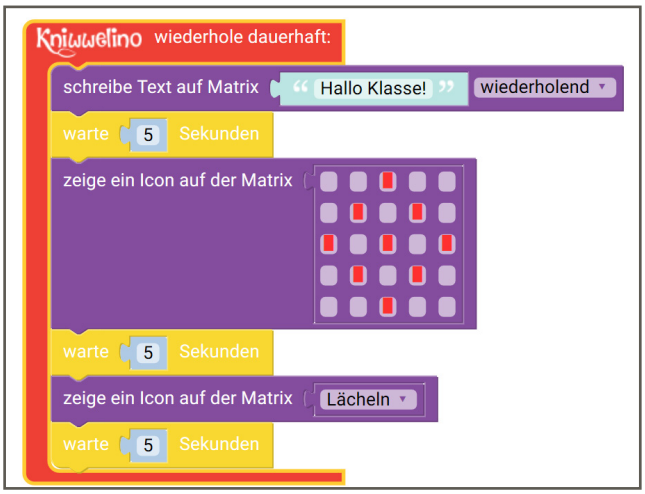


- Anzeigedauer verändern.

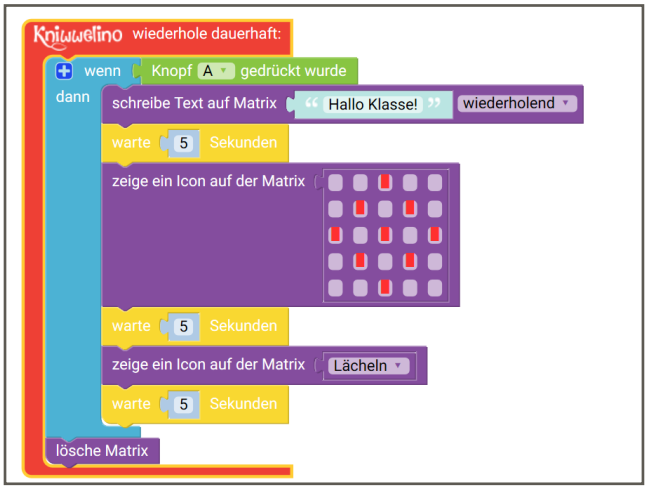
- Anzeige der Matrix verändern.



- Mehrere Befehle *(Matrix)* nacheinander eingeben und die Anzeigedauer variieren.



In einer zweiten Phase können Grundbegriffe des Computational Thinking eingeführt werden, wie zum Beispiel Loops *(Schleifen)* oder Logik *(Wenn... dann)*.



Die SchülerInnen sollen zu einem Lesetext eine passende Nachricht *(als Text, Bild oder Icon)* anzeigen lassen, die ihre Meinung zum Text oder den Hauptfiguren ausdrückt. Sie sollen die Wahl ihrer Anzeige begründen.

Vertiefung:

Die SchülerInnen definieren zwei Zustände und rufen diese durch das Drücken der Knöpfe A und B auf.

Tipps und Tricks:

Anhand der Matrixanzeige können die SchülerInnen auch Gefühle oder Stimmungen ausdrücken. Dies könnte als Anlass zu einem Klassengespräch dienen.

Textnachrichten-Armband entwerfen/

Material

- Kniwwelino
- Tablet oder Computer
- Steckdosenleiste, Ladekabel oder Powerbank

Bezug zum Lehrplan

L'expression créatrice, l'éveil à l'esthétique et à la culture dans le domaine des arts plastiques : Imaginer et créer des œuvres bi- et tridimensionnelles

Bezug zum Lebensalltag

Handys und Smartphones gehören seit vielen Jahren zum Alltag. Die Lehrperson geht kurz auf die Entstehungsgeschichte der mobilen Telefone ein und zeigt ihre Entwicklung bis hin zum Smartphone auf.

Die Aufgabe "Kniwwelino Smartphone Text-nachrichten“ erlaubt es, einen Einblick in die Anfangstage des mobilen Telefons zu erlangen. Die SchülerInnen stellen ein einfaches, funktionsfähiges Armband her, das ihnen erlaubt, über Textnachrichten zu kommunizieren.

Ablauf

Die SchülerInnen verbinden ihr Kniwwelino mit dem WLAN und öffnen im Browser die Seite http://code.kniwwelino.lu.

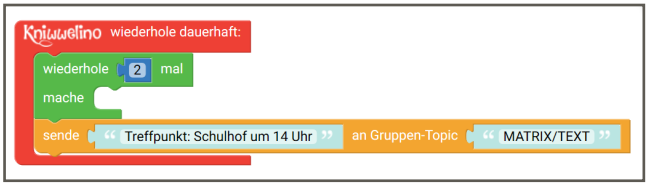
Da vorwiegend Elemente aus dem Menü "Nachrichten" verwendet werden, schieben die SchülerInnen die folgenden Blöcke in den Arbeitsbereich:

Damit das Versenden der Textnachrichten klappt, müssen alle Kniwwelinos in einer Gruppe angemeldet sein. Alle Nachrichten werden auf diese Weise an die gesamte Gruppe weitergeleitet. In diesem Beispiel ist es die Gruppe "Meine_Klasse".

Kleine Textnachrichten werden in den vorgesehenen "Sende"-Block eingetragen.

Gegebenenfalls kann zusätzlich bestimmt werden, dass die Nachricht zweimal gesendet wird.

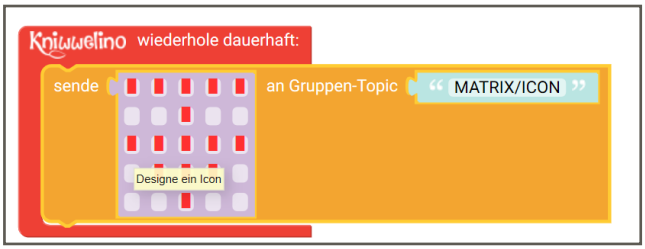
Dieser Block befindet sich im Menü "Schleifen“.



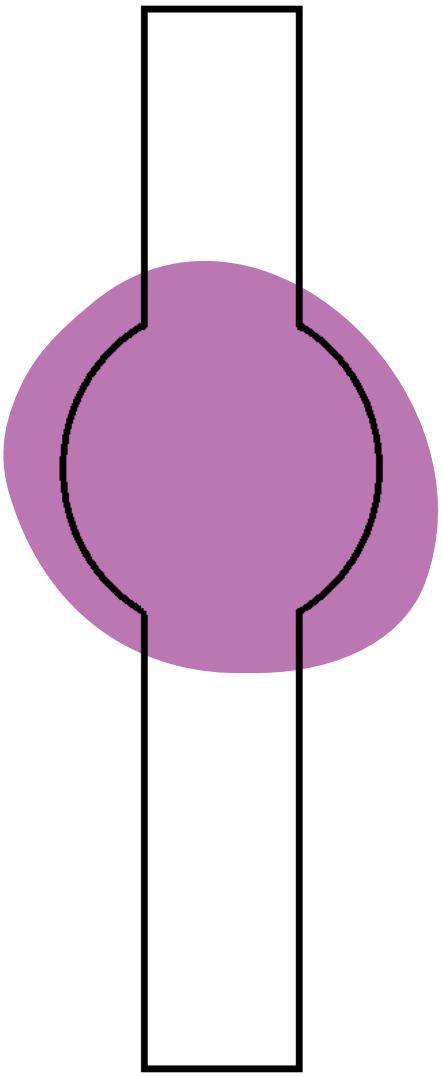
Sind Gruppe und Nachricht in den Blöcken definiert, muss das Programm nur noch mit einem Klick auf die Kniwwelinos übertragen werden.

Nach etwa 30 Sekunden wird die Nachricht "Treffpunkt: Schulhof um 14 Uhr“ auf allen Kniwwelinos, die in der Gruppe angemeldet sind, angezeigt.

Zusätzlich zu den Textnachrichten können auch Smileys verschickt werden. Hierzu können die SchülerInnen auf einer 5x5-Punktematrix ihre Smileys entwerfen und dann in die Blöcke "Matrix/Icon" *(unter Nachrichten)* einsetzen.



Die SchülerInnen gestalten ein individuelles Armband für das Kniwwelino-Board.



"Quelle: https://doku.kniwwelino.lu/en/instructions/wristband"

Tipps und Tricks:

Die erstellten Programme werden jedes Mal gespeichert, um auch später darauf zugreifen zu können. Teile eines Programmes können dadurch kopiert und in ein neues Programm integriert werden.

Wichtig: Erst nach dem Hochladen des neuen Codes wird dieser auf der Matrix des Boards angezeigt. Dieser Vorgang kann bis zu einer Minute dauern.

Stein-Papier-Schere/

Material

- Kniwwelino
- USB Kabel *(+ Powerbank)*
- Tablet oder Computer
- 1 Blatt Wellpappe oder glatte Pappe
- Schere & Bastelmesser
- Kleber

Bezug zum Lehrplan

Mathématiques : Résolution de problèmes d'arithmétique : Interpréter et évaluer les informations et les résultats

Bezug zum Lebensalltag

In dieser Unterrichtseinheit wird das Spiel Stein-Papier-Schere digital umgesetzt. Die SchülerInnen können später gegen ihre Freunde oder den Computer antreten.

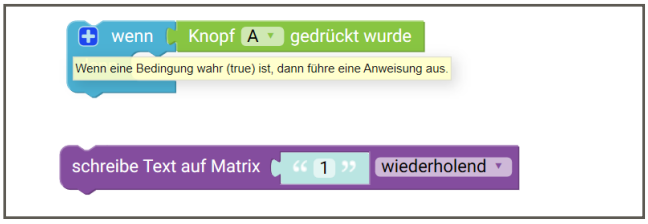
Computer ermitteln oft Zufallszahlen oder –werte. Überlege, wo das sein könnte.

Computer sind Rechner. Zufall ist nicht berechenbar. Es können Wahrscheinlichkeiten berechnet werden, doch wie lassen sich Zufallszahlen "errechnen"? Die Antwort auf diese Frage ist eine Herausforderung.

Ablauf

Für diese Aktivität werden eine Reihe von "Wenn... dann" Blöcken aus dem Bereich "Logik" verwendet. Du kannst dein Spiel schrittweise programmieren. Verwende die angezeigten Blöcke und setze sie in die richtige Reihenfolge. Nach jedem Schritt kannst du dein Programm testen, indem du auf ▶ klickst.

Schritt 1: Lass auf Knopfdruck eine Zahl erscheinen. Du wirst deinem Kniwwelino sagen: „Wenn ich Knopf A drücke, dann schreibe "1" auf die Matrix". Verwende dazu folgende Blöcke:



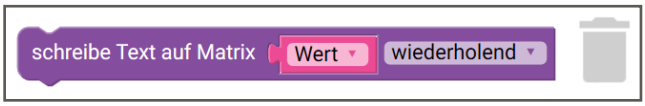
Schritt 2: Lass auf Knopfdruck eine zufällige Zahl zwischen 1 und 3 anzeigen.

Du wirst deinem Kniwwelino sagen: „Wenn ich Knopf A drücke, dann wähle eine zufällige Zahl zwischen 1 und 3 und schreibe diese Zahl in die Variable *Wert*. Verwende folgende Bausteine:

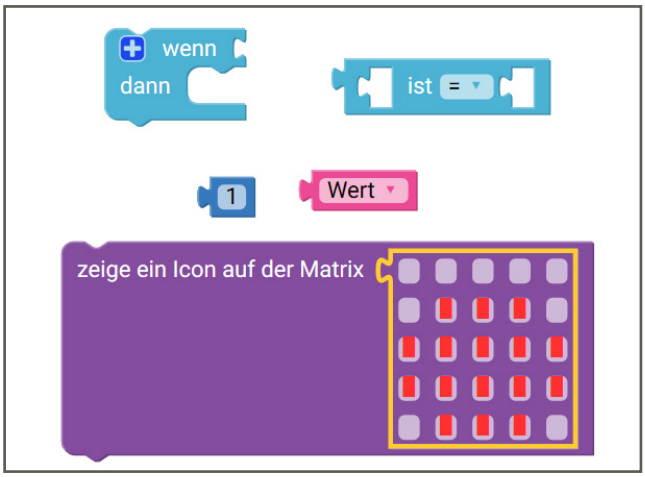
Schritt 3: Zeige unterschiedliche Bilder für die Zahlen 1, 2 und 3.

Anstatt die Zahl in *Wert* anzuzeigen, sagst du nun: *„Wenn die Zahl Wert gleich 1 ist, dann zeige ein Bild an (zum Beispiel eine 'Schere'). Wenn die Zahl Wert gleich 2 ist, dann zeige ein anderes Bild an (zum Beispiel einen 'Stein'), Wenn die Zahl Wert gleich 3 ist, dann zeige ein drittes Bild an (zum Beispiel ein 'Papier')."*

Entferne den Block, der die Zahl anzeigt.



Ersetze ihn mit deiner ersten Bedingung aus folgenden Blöcken.



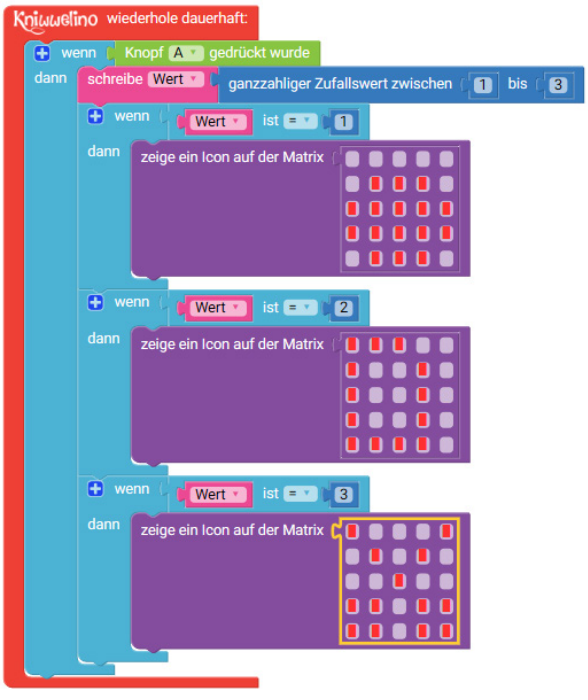
Wiederhole diese Schritte für die beiden anderen Bedingungen

Vertiefung:

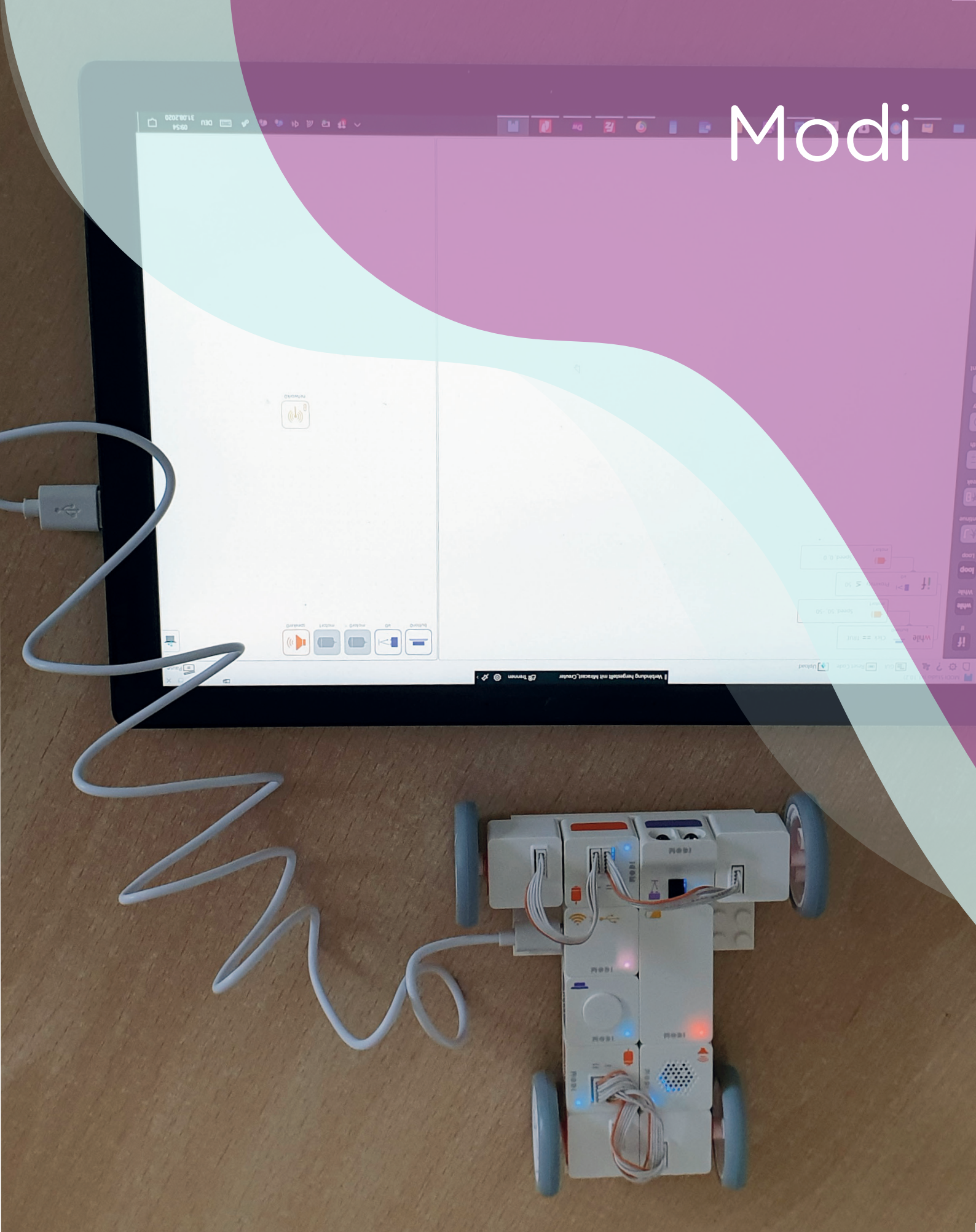
Bastle passende Hände für dein Kniwwelino und duelliere dich mit deinen FreundInnen.

* *"Quelle: https://doku.kniwwelino.lu/de/instructions/rockpaperscissors*

Mögliche Lösung:



Modi



Taschenlampe/

Material

- Modi-Studio-App
- Modi-Netzwerk-Modul
- 2 Legosteine, um die Elemente zu verbinden *(optional)*

Bezug zum Lehrplan

Mathématiques : Résolution de problèmes d'arithmétique : Résoudre un problème d'arithmétique

Bezug zum Lebensalltag

Baue deine eigene digitale Taschenlampe *(mit Dimmer)*, die du auch als Morsegerät einsetzen kannst.

Ablauf

Interessant ist es, bereits von Anfang an zwei Lichtstärken vorzusehen, je nachdem, wie der Drehknopf eingestellt ist *(-> Dimmer)*. Mit Hilfe der "if" und "else if" Funktion gelingt dies relativ leicht.

Vertiefung:

Mit Hilfe ihrer Taschenlampe verschicken die SchülerInnen geheime Botschaften. Hierfür setzen sie sich mit den Morsezeichen auseinander. Die SchülerInnen überlegen sich eine kurze Botschaft, die sie den KlassenkameradInnen zuschicken.

→

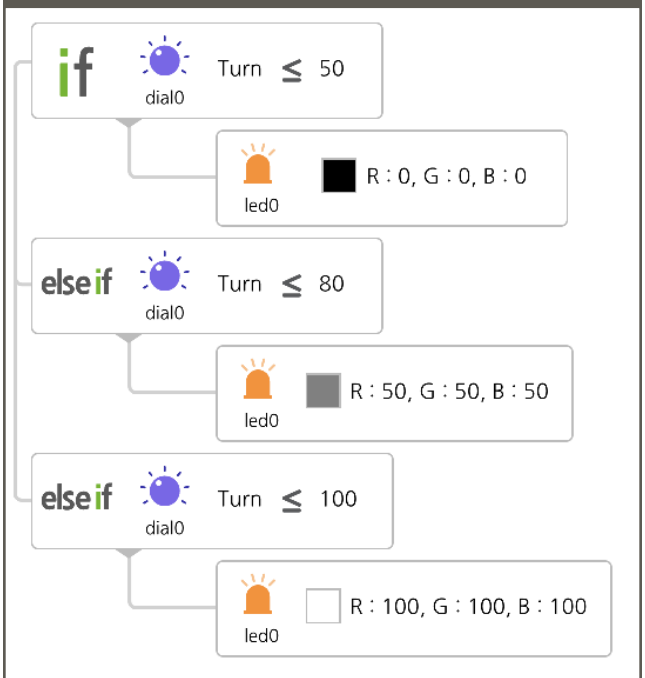
Dieser/diese soll dann mit Hilfe der Codetabelle die Botschaft entschlüsseln.

(Link zur Standard-Codetabelle: https://de.wikipedia.org/wiki/Morsezeichen)

Tipps und Tricks:

Unter folgendem Link gibt es ein passendes Erklärvideo zu dieser Aktivität: https://www.youtube.com/watch?v=l8k9oa7sErY

Mögliche Lösung:



Geräuschpegel/

Material

- Batterie
- Modi-Studio-App
- Modi-Netzwerk-Modul
- Modi-Lampe-Modul
- Modi-Mikrofon-Modul

Bezug zum Lehrplan

Mathématiques : Grandeurs et mesures : Disposer d'une représentation mentale des grandeurs.

Bezug zum Lebensalltag

Computer und Roboter verfügen oft über Sensoren. Diese sind ihre Sinne, mit denen sie sehen, hören, fühlen... können. Im Gegensatz zu uns Menschen, erfassen die Sensoren Informationen viel präziser.

Lärm wird in Dezibel gemessen. Die Einheit ist das Bel, ein Dezibel entspricht einem Zehntel-Bel.

Ablauf

Oft werden Behauptungen aufgestellt, die ein Gefühl und nicht unbedingt die Realität wiedergeben, zum Beispiel bei der Wahrnehmung von Lärm und Geräuschen. In dieser Einheit wird Klarheit darüber geschaffen, wie oft der Lärmpegel während einer Schulstunde oder eines Schultages ein gewisses Niveau übersteigt. Die SchülerInnen programmieren den Modi so, dass die Lampe aufleuchtet, wenn der Lärmpegel über 50 Dezibel steigt. Jedes Mal, wenn der Lärmpegel den festgelegten Wert überschreitet, leuchtet die Lampe rot. Die SchülerInnen führen Statistik, wie oft und zu welchen Zeiten dies der Fall ist. Eventuell sind hier nach mehrmaligem Durchführen bestimmte Muster zu erkennen, die später in einem Klassengespräch diskutiert werden können.

Vertiefung:

Zusätzlich zur Lampe kann mit dem Buzzer auch ein akustisches Signal gegeben werden, wenn der Geräuschpegel in der Klasse zu hoch ist.

Tipps und Tricks:

Es ist wichtig, nach dem Befehlsblock zum Aufleuchten der LED stets einen Warteblock einzusetzen. Wenn die verschiedenen Leuchtzustände der LED zu schnell wechseln, ist der Effekt nicht erkennbar ist.

Mögliche Lösung:

Polizei/

Material

- Batterie
- Modi-Knopf-Modul
- 2 Modi-Lampen-Module
- Modi-Netzwerk-Modul

Bezug zum Lehrplan

Mathématiques : Grandeurs et mesures : Disposer d'une représentation mentale des grandeurs

Bezug zum Lebensalltag

Polizeiwagen, Krankenwagen oder Autos vom Zoll benutzen unterschiedliche Lichtanlagen, um schnell und sicher durch den Verkehr zu kommen.

Ablauf

Die Polizei braucht eine neue Lichtanlage für ihre Autos. Die einzelnen Lampen sollen jedoch nicht beliebig an- und ausgehen, sondern einem ganz bestimmten Rhythmus folgen. Bevor die neuen Lichtanlagen in Produktion gehen, wird erprobt, welcher Rhythmus am besten bei den Kindern in Luxemburg ankommt. Zusätzlich soll die Lichtanlage mit einem einfachen Drehknopf ein- und ausgeschaltet werden können.

Die SchülerInnen probieren verschiedene Kombinationen aus und dokumentieren diese genau, damit sie später wieder abrufbar sind. Welche Kombination bevorzugen die Kinder?

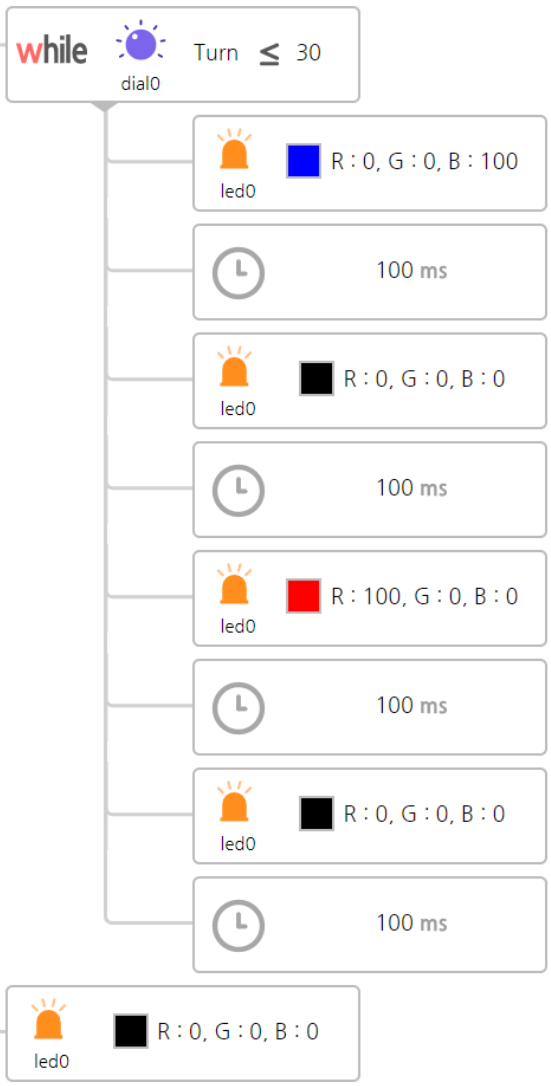
Vertiefung:

- Die SchülerInnen suchen sich eigene Farbreihenfolgen aus und programmieren sie.
- Der Drehknopf wird durch einen einfachen Knopf ersetzt und die Programmierung angepasst.
- Zusätzlich zum Lichtsignal wird ein Tonsignal umgesetzt.

Tipps und Tricks:

RGB steht für Red, Green, Blue *(also rot, grün und blau)*. Die LED kann in vielen Farben aufleuchten. Die Farben werden jeweils aus verschiedenen Anteilen der Farben Rot, Grün und Blau zusammengemischt. Die meisten Bildschirme funktionieren nach diesem Prinzip.

Mögliche Lösung für eine einfache Lichtanlage:



Mögliche Lösung für eine Lichtanlage mit Knopf und Tonsignal:

Instrument/

Material

- Batterie
- Modi-Netzwerk-Modul
- Modi-Drehknopf-Modul
- Modi-Lautsprecher-Modul
- Modi-IR*(Infrarot)*-Modul
- Modi-Display-Modul

Bezug zum Lehrplan

L'expression créative, l'éveil à l'esthétique et à la culture dans le domaine de la musique : Percevoir la musique

Bezug zum Lebensalltag

In dieser Einheit lernen die SchülerInnen, wie eine Lichtharfe funktioniert. Zur Einführung schauen sie sich ein Video an, wie eine Lichtharfe gespielt wird.

Die Einheit eignet sich ebenfalls, um über Tonfrequenzen zu sprechen. Im Durchschnitt hört der Mensch Frequenzen zwischen 20 und 20 000 Hertz. Tonhöhen werden durch die Anzahl der Schwingungen der Schallwellen pro Sekunde, der Frequenz, bestimmt. Die Frequenz wird in Hertz *(Hz)* gemessen.

Ablauf

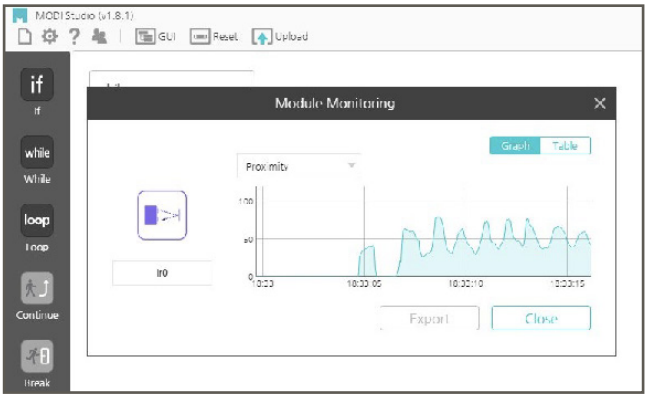
In der Klasse wird ein Instrument gebaut, das es ermöglicht, verschiedene Töne zu spielen, ohne das Instrument zu berühren.

Die SchülerInnen überlegen, welche Blöcke dazu gebraucht werden, und begründen ihre Wahl.

Die benötigten Blöcke sind die Batterie, das IR-Modul und der Lautsprecher.

Das IR-Modul ermöglicht Werte zu ändern, ohne das Instrument *(die Blöcke)* zu berühren.

Zuerst schließen die SchülerInnen das IR-Modul an, doppelklicken auf die Ikone und sehen sich im Diagramm an, wie sich die Werte verändern, wenn sie die Hand näher heran oder weiter weg bewegen.



Im Anschluss wird in der Klasse besprochen, dass hohe Töne höhere Frequenzen haben und tiefere Töne niedrigere Frequenzen. Wird die Frequenz des Lautsprechers anhand der Werte des IR-Moduls definiert, lassen sich die Höhe der Töne beeinflussen.

Nun schließen die SchülerInnen auch den Lautsprecher an und setzen die Frequenz des Lautsprechers mit dem Wert des IR-Moduls gleich. Die Lautstärke kann frei gewählt werden.

Das Instrument ist fertig. Mit den Bewegungen der Hand vor dem IR-Modul können verschiedene Töne gespielt werden .

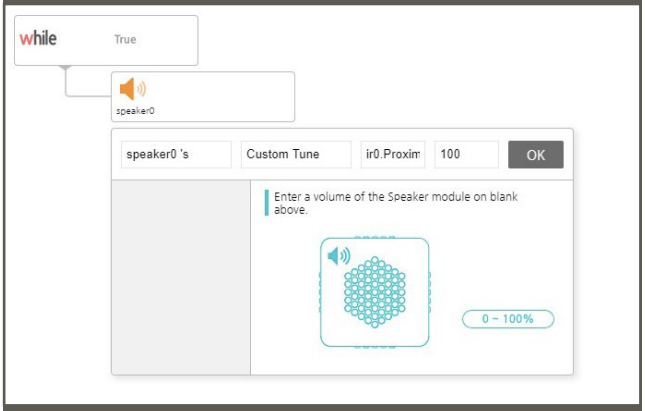
Vertiefung:

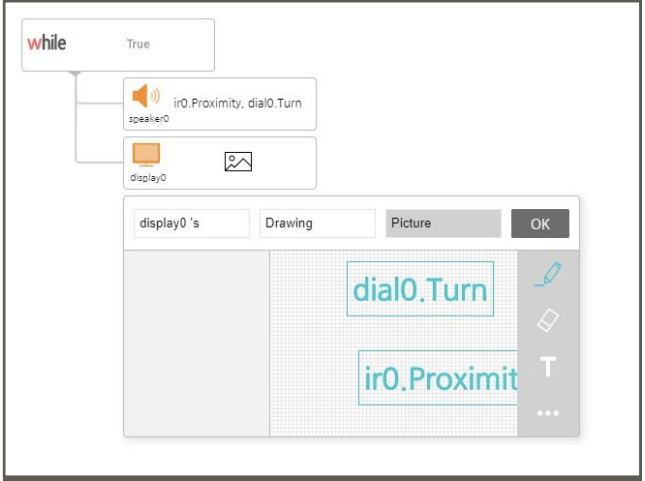
Die SchülerInnen ordnen vor dem IR-Modul mit Hilfe von Strichen *(wie bei der Gitarre)* den Stellen bestimmte Töne zu und spielen Melodien.

Zusätzlich wird ein Drehknopf hinzugefügt, der es erlaubt, die Lautstärke durch Drehen zu verändern.

Mit dem Display lassen sich die Werte des IR-Moduls und des Drehknopfes anzeigen.

Mögliche Lösung:





Rover/

Material

- Batterie
- Modi-Netzwerk-Modul
- Modi-Lautsprecher-Modul
- Modi-Knopf-Modul
- Modi-Motormodul mit 2 Motoren
- Modi-IR*(Infrarot)*-Modul
- Legosteine

Bezug zum Lehrplan

Mathématiques : Grandeurs et mesures : Disposer d'une représentation mentale des grandeurs

Bezug zum Lebensalltag

Distanzen werden in Längeneinheiten angegeben: mm, cm, dm, m... Zum Messen einer Distanz eignet sich ein Lineal oder ein Metermaß. Dafür braucht es weder Strom noch einen Computer.

In abgelegenen Regionen der Erde oder auf anderen Planeten ist es oft nicht möglich, mit einem Metermaß Distanzen zu messen. Für diesen Zweck setzen die ForscherInnen Maschinen ein, die mit speziellen Sensoren ausgestattet sind. Zum Bestimmen einer Distanz kann ein Infrarotsensor eingesetzt werden. Das IR-Modul bestimmt die Distanz zu einem Objekt, indem es den reflektierenden Wert des Infrarotlichts misst, welches vom Modul ausgestrahlt wurde.

Ablauf

In dieser Unterrichtseinheit bauen die SchülerInnen einen Rover *(Roboter)*, der fähig ist, von einem Punkt zum anderen zu fahren, mögliche Hindernisse zu erkennen und diese mit einem Tonsignal zu signalisieren. Der Rover wird per Knopfdruck gestartet.

Die SchülerInnen überlegen, welche Blöcke dazu gebraucht werden. Diese legen sie vor sich hin und begründen ihre Wahl.

Die gebrauchten Blöcke sind die Batterie, das Motormodul mit 2 Motoren, der IR-Sensor und der Lautsprecher. Der IR-Sensor erlaubt es den Roboter so zu programmieren, dass er bei einer gewissen Distanz zu einem Objekt stoppt.

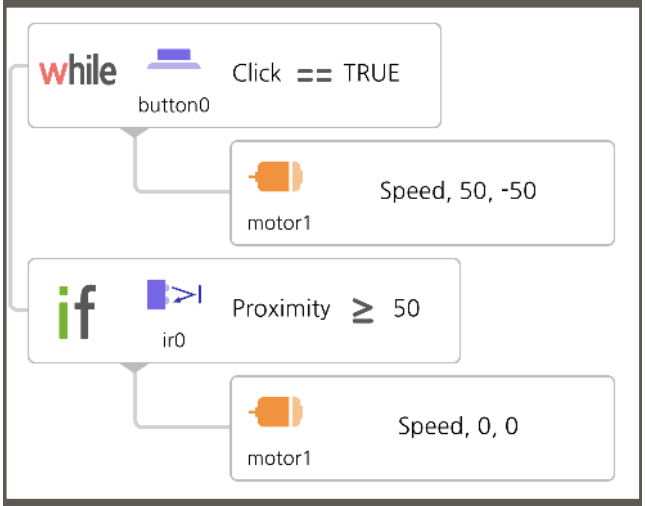
Zuerst schließen die SchülerInnen den IR-Sensor an, doppelklicken auf die Ikone und sehen sich im Diagramm an, wie die Werte ändern, wenn sie ein Objekt näher heran oder weiter weg bewegen. In einer Tabelle halten die SchülerInnen fest, welcher Wert mit welcher Distanz übereinstimmt. Anschließend beginnen sie mit der eigentlichen Programmierung ihres Roboters. Es bietet sich an, zuerst mit einem Prototyp zu starten.

1. *Der Rover soll von einer Seite des Tisches zur anderen rollen können.*
2. *Der Rover soll 4 Sekunden rollen und dann stoppen.*
3. *Der Rover soll auf Knopfdruck starten und nach 4 Sekunden stoppen.*
4. *Der Rover soll auf Knopfdruck starten, auf seinem Weg zur anderen Seite des Tisches ein Hindernis erkennen und auf sicherer Distanz zu diesem zum Stehen kommen.*

Vertiefung:

Zusätzlich zum einfachen Stoppen lässt sich ein akustisches Geräusch einbauen, welches signalisiert, dass der Rover einen Gegenstand entdeckt hat.

Mögliche Lösung:



LOGIC GAME
Map 1
0
1
2

Quellen/

Fadel, C., Bialik, M., Trilling, B. & Schleicher, A. (2017). Die vier Dimensionen der Bildung: Was Schülerinnen und Schüler im 21. Jahrhundert lernen müssen *(J. Muuß-Merholz, Übers.)*. ZLL21 - der Verlag - Zentralstelle für Lernen und Lehren im 21. Jahrhundert e.V., Hamburg, www.ZLL21.de

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., New York

Resnick, M. (2017). Lifelong Kindergarden: Cultivating Creativity through Projects, Passion, Peers, and Play [Kindle-Version]. The MIT Press.

Reuter, C. (2020). Computational Thinking in der Grundschule, Masterarbeit (MA). Universitätslehrgang Game Based Media & Education. Fakultät für Bildung, Medien und Kommunikation. Zentrum für Angewandte Spielforschung. Donau Universität Krems, Krems

Wing, J. (2010). Research Notebook: Computational Thinking-What and Why? Carnegie Mellon School of Computer Science. https://www.cs.cmu.edu/link/researchnotebook-computational-thinking-what-and-why

Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. Educational Research Review, 22, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Weiterführende Unterrichtsideen/

Die Unterrichtsideen in diesem Heft stammen teilweise aus dem Internet und wurden an den Luxemburger Lehrplan angepasst.

Folgend einige Websites mit weiteren Unterrichtsideen:

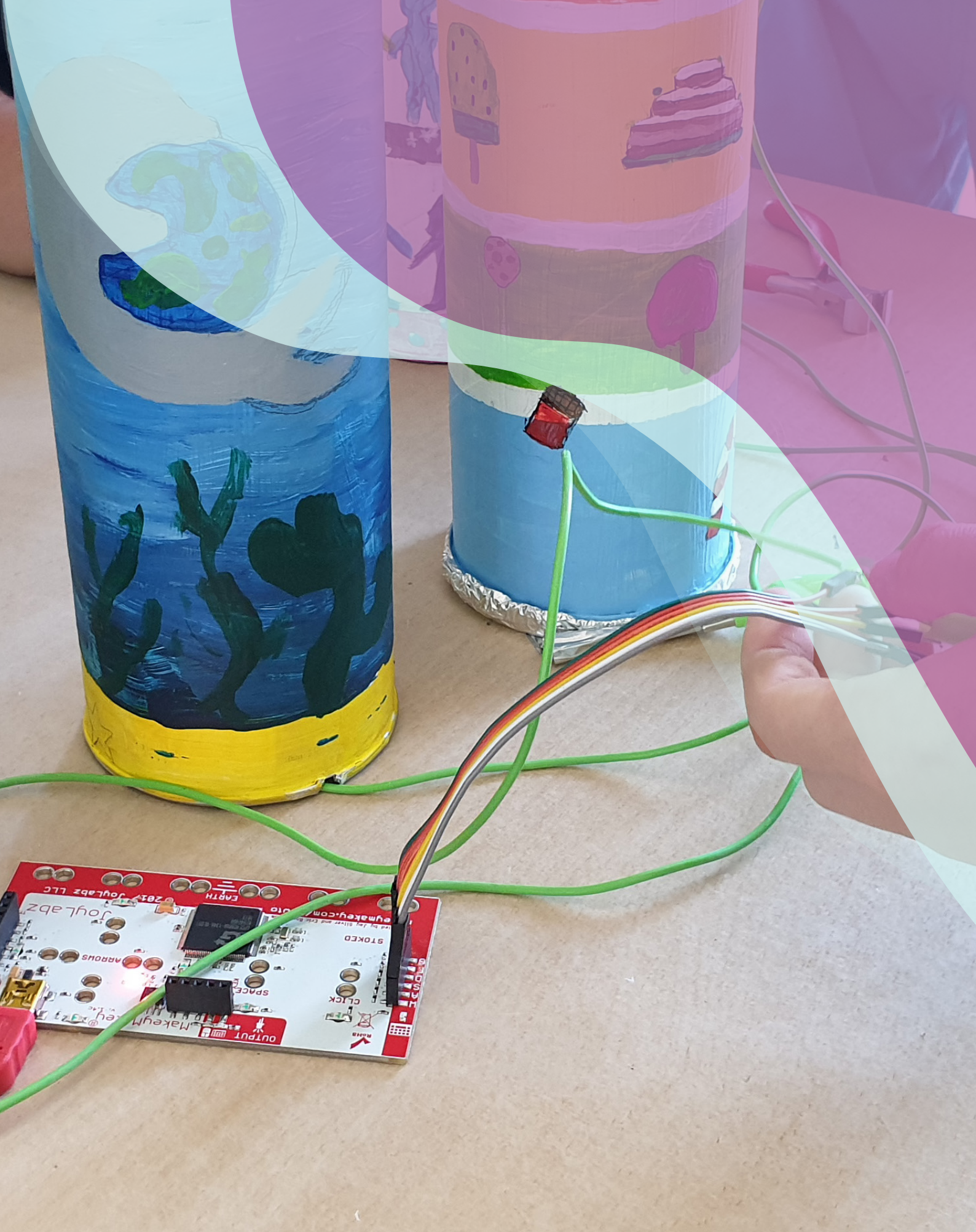
Unplugged
https://csunplugged.org/en/

Ozobot:
https://ozobot.com/educate

Scratch:
https://scratch.mit.edu/

Kniwwelino:
https://doku.kniwwelino.lu/en/start

LuxRobo Modi:
https://de.luxrobo.co.uk/


EARTH
STOKED
CLICK
OUTPUT

SCRIPT
Service de Coordination de la Recherche
et de l'Innovation pédagogiques et technologiques

SCRIPT, Service de Coordination de
la Recherche et de l'Innovation pédagogiques
et technologiques

eduPôle Walferdange
28, route de Diekirch
L-7220 Walferdange
Tél. : 247-85187
secretariat@script.lu
www.script.lu

SCRIPT, Walferdange 2020

ISBN 978-99959-1-204-8